



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Anàlisi de l'activitat dels desenvolupadors en plataformes de suport al desenvolupament col·laboratiu

Memòria del projecte

Autor: Miquel Xamaní Moreno

Director: Carles Farré Tost

Codirector: David Ameller

Professora de GEP: Olga Pons Peregort

Especialitat: Enginyeria Software

Data: 21/06/2017

Resum

Aquest treball de final de grau es centra en el disseny i desenvolupament d'un plugin per una de les plataformes de suport al desenvolupament col·laboratiu (p. ex. Slack, Trello, JIRA, etc.). El propòsit del plugin és extreure les dades de l'activitat dels desenvolupadors que realitzen en aquestes plataformes de manera automàtica i proveir Replan (una eina de planificació de *releases*) d'aquesta informació. A banda d'això, aquest treball inclou l'estudi d'aquestes plataformes que té per objectiu seleccionar-ne una tenint en compte les possibilitats tècniques que ofereixen i la viabilitat per obtenir les dades.

Abstract

This bachelor degree project focuses on the design and development of a plugin for one of support platform for collaborative development (e.g. Slack, Trello, JIRA, etc.). The purpose of this plugin is to extract data from developers activity on these platforms automatically and provide Replan tool (a release planning tool) this information. Apart from this, this project includes a study of these support platforms that aims to select one based on the technical possibilities they offer and the feasibility of obtaining the data.

Índex

1. Context	1
1.1. Introducció	1
1.2. Stakeholders	1
2. Estat de l'art	3
2.1. Contextualització	3
2.2. Plataformes	4
2.2.1. Slack	5
2.2.2. Trello	6
3. Abast	7
3.1. Definició de l'abast	7
3.2. Obstacles i riscos	8
4. Metodologia i rigor	9
4.1. Mètodes de treball	9
4.2. Eines de seguiment	10
4.3. Mètode de validació	10
5. Planificació temporal	11
5.1. Planificació general	11
5.1.1. Durada estimada del projecte	11
5.1.2. Recursos	11
5.1.3. Valoració d'alternatives i pla d'acció	12
5.1.4. Consideracions globals	13
5.2. Descripció de les tasques	13
5.2.1. Estudi previ	13
5.2.2. Anàlisi i especificació de requisits	13
5.2.3. Construcció 1 (comunicació Replan-Sistema-Plataforma)	14
5.2.4. Construcció 2 (comunicació Plataforma-Sistema-Replan)	14
5.2.5. Elaboració de la memòria	15
5.3. Desviacions en la planificació	15
5.4. Canvis en la planificació	16
6. Gestió econòmica	17
6.1. Estimació de costos	17
6.1.1. Costos directes per activitat	17
6.1.2. Costos indirectes	19
6.1.3. Contingència	19
6.1.4. Imprevistos	19
6.1.5. Cost total	20
6.2. Control de la gestió	20
7. Elecció de la plataforma de suport	22
7.1. Possibles solucions	22
7.1.1. Aplicació de Slack	22

7.1.1.1. Configuració inicial	23
7.1.1.2. Funcionament	23
7.1.1.3. Valoració de la solució	24
7.1.2. Trello com a consumidor i notificador de canvis en l'estat de les tasques del Replan	24
7.1.2.1. Configuració inicial	25
7.1.2.2. Funcionament	26
7.1.2.3. Valoració de la solució	26
7.1.3. Trello com a interfície del Replan	26
7.1.3.1. Configuració inicial	27
7.1.3.2. Funcionament	27
7.1.3.3. Valoració de la solució	27
7.2. Justificació de la solució escollida	28
8. Representació de la planificació a Trello	29
8.1. Elements de Trello	29
8.1.1. Tauler	29
8.1.2. Targeta	30
8.1.3. Llista	31
8.2. Replanificació	32
9. Especificació del sistema	34
9.1. Requisits funcionals	34
9.1.1. Casos d'ús aplicació web	35
9.1.1.1. Autoritzar plugin	35
9.1.1.2. Inicialitzar tauler	36
9.1.1.3. Consultar canvis estat	39
9.1.1.4. Finalitzar tasca	40
9.1.1.5. Rebutjar tasca	40
9.1.1.6. Consultar canvis estat tasca	41
9.1.1.7. Actualitzar tauler	41
9.1.2. Casos d'ús Trello	42
9.1.2.1. Modificar estat tasca	42
9.2. Requisits de qualitat	43
9.2.1. Requisits de percepció	43
9.2.2. Requisits d'usabilitat i humanitat	44
9.2.3. Requisits de rendiment	45
9.2.4. Requisits operacionals	46
9.2.5. Requisits de manteniment i suport	46
9.3. Model conceptual	47
9.3.1. Replan	47
9.3.2. Trello	48
9.3.3. Plugin i relacions entre sistemes	49
10. Desenvolupament del sistema	52
10.1. Front-end del Plugin	52
10.1.1. Diagrama de navegabilitat	54

10.1.2. Diagrama disseny intern	57
10.2. Back-end del Plugin	60
10.2.1. Diagrama classes back-end	61
10.2.1.1 Diagrames classes capa controladors	62
10.2.2.2 Diagrama classes capa de serveis	65
10.2.2.3 Diagrames classes capa de domini	66
10.2.2.4 Diagrama interacció entre capes de servidor	71
10.2.2.5 Classes auxiliars	72
11. Implementació	73
12. Testing	75
12.1. Creació del tauler	75
12.2. Reaccionar quan es mou una targeta de llista	76
12.3. Actualització del tauler	79
13. Desplegament	87
14. Sostenibilitat i compromís social	88
14.1. Dimensió ambiental	88
14.2. Dimensió econòmica	88
14.3. Dimensió social	89
15. Conclusions	90
15.1. Assoliment dels objectius	90
15.2. Coneixements aplicats i adquirits	90
15.3. Treball futur	91
16. Referències	92
Annex A: Diagrames de Gantt planificació inicial	94
Annex B: Diagrames de Gantt final	95
Annex C: Especificació de l'API	96
Annex D: Planificació en format JSON 1	127
Annex E: Planificació en format JSON 2	134

Índex de figures

Figura 1: Replan architecture	4
Figura 2: Arquitectura dels sistemes que intervenen	22
Figura 3: Planificació d'una release a Trello (moment inicial)	29
Figura 4: Detall d'una targeta de Trello	30
Figura 5: Diagrama de casos d'ús aplicació web	34
Figura 6: Diagrama casos d'ús Trello	42
Figura 7: Diagrama conceptual sistema Replan	47
Figura 8: Diagrama model conceptual sistema Trello	48
Figura 9: Diagrama model conceptual Plugin associació de recursos	49
Figura 10: Diagrama model conceptual relació Board-Plan	50
Figura 11: Diagrama model conceptual relació Log-Action	50
Figura 12: Diagrama model conceptual FeatureCard-Card	51
Figura 13: Components bàscis d'una aplicació d'AngularJS	53
Figura 14: Diagrama de navegabilitat (login, registre i pàgina home)	54
Figura 15: Diagrama de navegabilitat (procés de carregar planificació a Trello 1a part)	55
Figura 16: Diagrama de navegabilitat (procés de carregar planificació a Trello 2a part)	56
Figura 17: Diagrama disseny intern (login i registre)	57
Figura 18: Diagrama disseny intern (refusar card i primeres 2 pàgines del procés ...)	58
Figura 19: Diagrama disseny intern (darreres 3 pàgines del procés ...)	59
Figura 20: Esquema general components aplicació similar al back-end del plugin	60
Figura 21: Diagrama de les capes del backend	61
Figura 22: Diagrama classes DTO 1	62
Figura 23: Diagrama classes DTO 2	63
Figura 24: Diagrama classes controladors	64
Figura 25: Diagrama classes serveis	65
Figura 26: Diagrama classes JPA	66
Figura 27: Diagrama classes JPA per simular Replan	67
Figura 28: Diagrama classes interfície d'accés a dades	68
Figura 29: Diagrama classe Domain controller	69
Figura 30: Diagrama interacció entre classes de la capa de domini	70
Figura 31: Diagrama interacció classes diferents capes del back-end	71
Figura 32: Diagrama classe enumeració	72
Figura 33: Diagrama classe auxiliar d'operacions	72
Figura 34: Exemple planificació Replan	75
Figura 35: Captura planificació carregada a Trello	76
Figura 36: Targetes mogudes a la llista In Progress	77
Figura 37: Captura pàgina home després d'haver mogut targetes a la llista In Progress	77
Figura 38: Targetes mogudes a la llista Done	78
Figura 39: Captura pàgina home després d'haver mogut targetes a la llista Done	78
Figura 40: Planificació utilitzada per replanificar	79
Figura 41: Planificació utilitzada per replanificar a Trello	80
Figura 42: Tauler abans de replanificar	81
Figura 43: Tauler després de replanificar (cas eliminació de targetes)	81

Figura 44: Tauler abans de replanificar (cas aparició de nova targeta)	82
Figura 45: Tauler després de replanificar (cas aparició de nova targeta)	83
Figura 46: Detall de la targeta afegida per la replanificació	84
Figura 47: Tauler abans de replanificar (cas canvis d'assignació)	85
Figura 48: Tauler després de replanificar (cas canvis d'assignació)	85
Figura 49: Detall de la targeta del canvi d'assignació	86
Figura 50: Diagrama de desplegament	87
Figura 51: Diagrama Gantt - Fases d'Estudi previ i Anàlisi i especificació de requisits	94
Figura 52: Diagrama Gantt - Fase de Construcció 1	94
Figura 53: Diagrama Gantt - Fase de Construcció 2	94
Figura 54: Diagrama Gantt - Fase de Construcció 1 després de les desviacions	95
Figura 55: Diagrama Gantt - Fases de Construcció 2 i Elaboració de la memòria...	96

Índex de taules

Taula 1: Metodologia de treball	9
Taula 2: Recursos materials	12
Taula 3: Rols implicats en el desenvolupament del projecte	17
Taula 4: Costos directes per activitat	18
Taula 5: Costos indirectes	19
Taula 6: Costos imprevistos	20
Taula 7: Cost total del projecte	20
Taula 8: Repercussió de les desviacions en el cost final del projecte	21
Taula 9: Equivalència de conceptes	25
Taula 10: Representació de la informació d'una tasca a Trello	25
Taula 11: Representació de la informació d'una tasca en una targeta de Trello	31
Taula 12: Requisit de qualitat aparença	43
Taula 13: Requisit de qualitat estil	43
Taula 14: Requisit de qualitat facilitat d'ús i aprenentatge	44
Taula 15: Requisit de qualitat comprensió i cortesia	44
Taula 16: Requisit de qualitat velocitat i latència	45
Taula 17: Requisit de qualitat robustesa i tolerància a errors	45
Taula 18: Requisit de qualitat interacció amb sistemes adjacents	46
Taula 19: Requisit de qualitat adaptabilitat	46
Taula 20: Matriu de sostenibilitat	88

1. Context

1.1. Introducció

Aquest projecte és un Treball Final de Grau de l'especialitat d'Enginyeria del *Software* de la Facultat d'Informàtica de Barcelona (Universitat Politècnica de Catalunya). Es tracta d'un projecte de modalitat A proposat pel departament d'Enginyeria de Serveis i Sistemes d'Informació (ESSI). Membres d'aquest departament en col·laboració amb grans empreses internacionals tecnològiques formen part del projecte europeu SUPERSEDE^[1]. En el marc d'aquest projecte s'ha desenvolupat Replan^[2], una eina de suport a la planificació del desenvolupament de projectes de *software*. Des de SUPERSEDE es vol millorar aquesta eina perquè sigui compatible amb **Continuous Software Release Planning** (CSRP)^[3]. La idea principal del CSRP consisteix a realitzar noves planificacions (re-planificacions) de la *release* actual cada vegada que l'activitat d'un desenvolupador provoca un canvi en la planificació actual (ex. S'ha completat la implementació d'un requisit abans de la data de finalització definida), necessitant una actualització de la planificació. Per tant, per tal que l'eina suporti el CSRP és imprescindible que pugui rebre el feedback de l'activitat dels desenvolupadors des de les plataformes de suport al desenvolupament col·laboratiu amb les quals treballen com Trello o Slack.

L'objectiu principal del projecte és oferir una solució *software* que permeti monitorar l'activitat que realitzen els desenvolupadors a les plataformes de suport al desenvolupament col·laboratiu més utilitzades i informar a l'eina Replan d'aquelles activitats que causen que sigui necessari una nova planificació de la *release* actual.

1.2. Stakeholders

Les parts interessades o *stakeholders* d'aquest projecte són totes aquelles persones o empreses que tenen alguna mena de relació amb el projecte ja sigui perquè intervenen en el desenvolupament del projecte o esperen beneficiar-se del resultat d'aquest.

1.2.1. Desenvolupador del projecte

Una de les principals parts interessades, ja que està implicat en tot el procés i resulta directament afectat per l'evolució i pel resultat obtingut d'aquest. En aquest cas, és un únic desenvolupador que també és l'autor d'aquest document. En tractar-se aquest projecte d'un TFG, el desenvolupador també se'n beneficiarà, ja que li permetrà acabar el grau i rebre el títol universitari.

1.2.2. Director i codirector

Professors del departament d'Enginyeria de Serveis i Sistemes d'Informació (ESSI) que dirigeixen el TFG que s'encarregaran de guiar i supervisar el treball realitzat pel desenvolupador del projecte.

1.2.3. Projecte SUPERSEDE

Com ja s'ha mencionat anteriorment, l'eina Replan ha estat desenvolupada per membres del projecte SUPERSEDE, entre els quals es troben el director i el codirector d'aquest TFG. Gràcies al resultat d'aquest projecte podran disposar del feedback provinent de l'activitat dels desenvolupadors en una plataforma de suport al desenvolupament col·laboratiu per utilitzar-lo a l'eina Replan podent així donar suport al CSRP.

1.2.4. Cap de projecte

Membre d'un equip de desenvolupament de *software* que s'encarreguen de la planificació de les *releases* d'un projecte i de revisar si les tasques implementades compleixen els requisits definits. És una de les principals parts interessades, ja que utilitzarà el *software* resultant del desenvolupament del projecte com suport a la seva feina de planificació i presa de decisions.

1.2.5. Desenvolupadors

Membre d'un equip de desenvolupament de *software* que realitzen les tasques tècniques. És una de les principals parts interessades, ja que a causa del *software* resultant del desenvolupament del projecte es veurà afectat per les planificacions generades i canviarà la seva manera d'utilitzar la plataforma de suport al desenvolupament col·laboratiu.

1.2.6. Empresa propietària d'una plataforma de suport al desenvolupament col·laboratiu

Empresa propietària d'una plataforma de suport al desenvolupament col·laboratiu per la qual es desenvoluparà el *software* permetent una integració amb l'eina Replan. L'empresa es veurà beneficiada perquè el *software* desenvolupat pot crear un efecte fidelitzant al només estar disponible per a aquesta plataforma.

2. Estat de l'art

2.1. Contextualització

Decidir què s'ha d'implementar, quan i per qui és una activitat que correspon a un cap de projecte i resulta crítica en el desenvolupament de *software*, aquesta activitat és coneguda com a **Software Release Planning** (SRP). Per tal de donar suport als caps de projecte a realitzar aquesta activitat, des del projecte SUPERSEDE s'ha desenvolupat Replan, una eina que realitza planificacions de *releases* de forma eficient tenint en compte les tasques a realitzar a la *release* i els recursos humans disponibles de l'empresa. A partir de les planificacions es pot saber quan finalitza cada tasca i quines tasques ha de realitzar cada recurs.

L'eina Replan està formada per 3 components:

- **Dashboard:** Una aplicació web que proporciona una interfície amb l'objectiu de facilitar l'accés a la principal funcionalitat de l'eina. Aquest component permet al cap de projecte proveir a l'eina la informació necessària per obtenir una planificació com els recursos humans disponibles, les tasques o funcionalitats a implementar i la duració de la *release*. A més a més de servir com a plataforma d'entrada de dades també permet visualitzar les planificacions generades per l'eina.
- **Controller:** Servei web encarregat de gestionar i emmagatzemar la representació interna dels elements del domini, com podria ser la informació relativa a les planificacions, recursos, *releases*, funcionalitats... El servei web disposa d'una API REST ben definida que proporciona al Dashboard tota la informació que necessita mostrar als usuaris i gestiona les peticions d'aquests com afegir un recurs a una *release* o obtenir una planificació de la *release*. Quan rep una petició d'obtenir una planificació d'una *release*, obté tota la informació necessària i l'envia a un últim component, Optimizer, que generarà la planificació i la retornarà. El Controller emmagatzemarà la planificació generada i la retornarà al Dashboard per ser visualitzada. El servei web també disposa d'una REST API addicional que permet que eines externes proveeixin l'eina d'una llista prioritzada de tasques a partir de la qual generar una planificació.
- **Optimizer:** Servei web que té com a únic objectiu generar planificacions de *releases*. Disposava d'una API que és consumida pel Controller amb l'objectiu de proporcionar entrada i sortida.

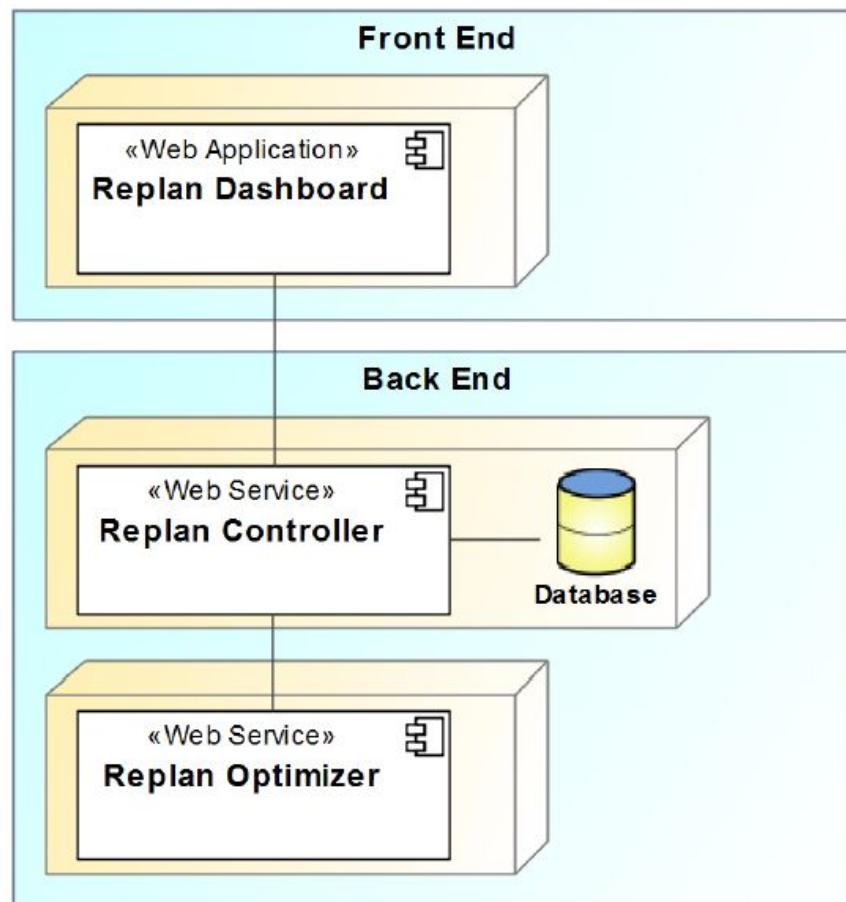


Fig 1. Ameller, D., Cassarino, A., Elvassore, V., Farré, C., Franch, X., Valerio, D. *Replan architecture* [Figura]. [Consultada: 22 Febrer 2017]. Disponible a: <
<http://www.essi.upc.edu/dameller/publications/ameller2017-saner-demo.pdf> >

L'eina Replan funciona correctament però es vol millorar perquè sigui compatible amb CSRP, ja que actualment no pot rebre el *feedback* de l'activitat dels desenvolupadors, informació que com ja s'ha mencionat és imprescindible per poder dur a terme CSRP. Els equips de desenvolupadors utilitzen plataformes de suport al desenvolupament col·laboratiu (com Slack o Trello, entre d'altres) en el seu entorn de treball per tal de realitzar un seguiment de la feina realitzada de forma organitzada. Com és en aquestes plataformes on es realitza, resulta de vital importància ser capaç de monitorar l'activitat que realitzen els desenvolupadors en aquestes plataformes per tal de proveir al Replan amb aquesta informació. D'aquesta manera s'obtidria el *feedback* que permet al Replan suportar el CSRP sent capaç de fer replanificacions d'una *release* en procés millorant així l'experiència que tenen els caps de projecte amb l'eina.

2.2. Plataformes

En aquest apartat s'introduiran les plataformes de suport al desenvolupament col·laboratiu escollides per avaluar la seva possible integració amb Replan. S'han escollit algunes de les plataformes més conegudes excloent JIRA^[4] a causa de la seva complexitat.

2.2.1. Slack

Slack^[5] és una plataforma bastant popular de comunicació entre els membres d'un equip de treball. Permet organitzar les conversacions pròpies d'un entorn de treball col·laboratiu en xats grupals anomenats canals. Gràcies als canals es poden organitzar les conversacions, ja que cada canal sol ser creat per parlar d'un tema en concret, com per exemple una entrega o una funcionalitat que s'està implementant. A través dels canals es poden compartir enllaços i arxius de forma molt ràpida i senzilla.

Slack també permet mantenir converses entre dos membres de l'equip a través dels missatges directes i permet la integració amb moltes altres eines a través de les aplicacions de Slack.

Slack ofereix la possibilitat de crear aplicacions que poden ser utilitzades per la resta d'equips de Slack i són fàcils d'instal·lar. Disposa d'una API documentada que ofereix moltes possibilitats perquè un desenvolupador pugui crear una integració amb Slack. Crear un *bot user* amb el que els usuaris poden conversar, configurar comandes per realitzar accions o fer arribar notificacions externes a Slack són les funcionalitats més destacables^[6].

2.2.2. Trello

Trello^[7] és una plataforma de gestió de projectes que permet organitzar les tasques d'un projecte en un tauler al qual se li pot posar un nom. El tauler conté llistes que permeten organitzar les targetes, és a dir les tasques, la qual cosa permet als usuaris representar la seva metodologia a Trello.

Les targetes contenen informació com el nom i la descripció, se'ls hi pot fixar una data de venciment, adjuntar arxius, es poden comentar i es poden assignar als membres del tauler.

Trello disposa d'una llista de complements que reben el nom de *power-ups*^[8] que permeten utilitzar algunes de les eines més conegudes a Trello com GitHub o el mateix Slack.

Trello ofereix als desenvolupadors la possibilitat de crear integracions dels seus sistemes amb la plataforma. Hi ha dos tipus d'integracions^[9]:

- **Power-up:** permet utilitzar uns elements GUI concrets proveïts per l'API de Trello per enriquir la interacció entre l'usuari i Trello. La publicació d'aquest tipus d'integracions perquè pugui ser compartit amb la resta d'equips depèn de l'aprovació de Trello, ja que apareixerà a la llista de *power-ups*.
- **Custom integration:** no té accés a la interfície de Trello però no requereix l'aprovació de Trello. Es pot compartir amb altres usuaris de manera externa a Trello.

Amb qualsevol tipus d'integració es té accés a les operacions de l'API de Trello que permeten crear, consultar, editar i eliminar taulers, llistes i targetes així com consultar informació dels usuaris i de l'equip. L'API de Trello també disposa de *webhooks*, que assignats a un element de Trello (tauler, llista o targeta) permeten notificar que ha tingut lloc una modificació d'aquest.

3. Abast

3.1. Definició de l'abast

El desenvolupament de la solució software constarà de diverses fases. La primera fase consistirà a analitzar l'arquitectura de l'eina Replan i la comunicació entre els diferents components. A continuació s'analitzaran les circumstàncies que poden causar un canvi en la planificació actual quedant aquesta obsoleta i per tant requerint una replanificació per part del Replan. Posteriorment s'analitzaran les possibilitats que ofereixen les plataformes mencionades amb anterioritat a l'hora de desenvolupar la solució software volguda. Finalment, tenint en compte la informació obtinguda dels passos anteriors, es pensaran possibles solucions, es valoraran i es justificarà l'elecció d'una. En aquest projecte només es triarà una plataforma tenint en compte la duració d'un TFG.

La segona fase consistirà a desenvolupar el sistema resultant format per 2 components ben diferenciats:

En primer lloc, una aplicació web orientada al cap de projecte que permetrà:

- Autoritzar el *software* a desenvolupar l'ús del seu compte de la plataforma.
- Triar quina planificació es vol visualitzar a la plataforma.
- Associar els recursos d'una planificació a un usuari de la plataforma i que es recordi pel pròxim ús de l'aplicació.
- Gestionar les tasques que han estat completades pels desenvolupadors decidint si aquestes es poden donar per finalitzades perquè compleixen els requisits definits i per tant cal una replanificació o per contra, no els compleixen rebutjant la tasca.
- Fer un seguiment de les etapes per les quals ha passat una tasca.

En segon lloc, un servei web que consumirà l'API d'una de les plataformes i que exposa una API per ser consumida tant pel Replan com per l'aplicació web que:

- Permetrà visualitzar planificacions generades per l'eina Replan a la plataforma.
- Rebre planificacions actualitzades (replanificacions) generades per l'eina Replan i permetre que es puguin visualitzar a la plataforma automàticament.
- Informarà els usuaris de retards en la finalització de les tasques d'una planificació.
- Crearà recordatoris de tasques de la planificació que estan a punt d'expirar i no s'han finalitzat encara.
- Detectar canvis en l'estat de les tasques de la planificació.
- Informar el Replan de circumstàncies que requereixen una replanificació (com la finalització abans d'hora d'una tasca o la no finalització d'una tasca abans de la data definida).

3.2. Obstacles i riscos

Els condicionants més destacables als quals s'han de fer front durant el desenvolupament del sistema s'han agrupat de la següent manera:

3.2.1. Restricció temporal

El Treball Final de Grau té una data límit fixada. Qualsevol mena de problema, inconvenient, *bug* o estancament durant la realització del mateix convindrà gestionar-lo adequadament per no endarrerir el projecte i, si s'escau, es replanificarà la feina a fer perquè no tingui repercussions majors.

3.2.2. Desconeixement de la tecnologia

L'avenç en el desenvolupament del projecte anirà estretament lligat amb el grau d'aprenentatge autònom que es vagi adquirint de Spring Boot, AngularJS i l'API de la plataforma escollida.

3.2.3. Retard en la implementació de la millora del Replan

Actualment s'està treballant en la millora del Replan perquè suporti CSRP i aquesta podria no trobar-se del tot acabada abans que finalitzi el TFG. Si es donés aquest cas, es podria resoldre creant una sèrie de jocs de proves que simularien el seu comportament.

3.2.4. Canvis en l'API de la plataforma

L'API de la plataforma escollida pot patir canvis que podrien arribar a causar problemes de compatibilitat amb tot aquell software que l'estigués consumint. Per solucionar aquest possible inconvenient caldrà estar al dia per poder detectar aquests canvis i reaccionar adaptant la implementació existent a aquesta nova situació.

4. Metodologia i rigor

4.1. Mètodes de treball

La metodologia de treball estarà dividida en 5 fases compostes de diferents tasques. La primera fase consistirà en un estudi previ del projecte. La segona fase serà l'anàlisi i especificació de requisits. La tercera i quarta fase són de construcció i es realitzaran seguint una metodologia iterativa i incremental^[10]. En elles es desenvoluparà l'aplicació web i el servei web a la vegada, ja que l'aplicació web depèn totalment de la informació proveïda pel servei web. A la tercera fase s'implementaran les funcionalitats imprescindibles per visualitzar les planificacions a la plataforma escollida. Per tant en aquesta fase s'estarà establint una comunicació en sentit Replan - Sistema a desenvolupar - Plataforma de suport. A la quarta fase s'implementarà la part que permet al Replan rebre *feedback* de l'estat de les tasques d'una planificació, i per tant sent compatible amb CSRP, completant així la comunicació en sentit invers i fent possible actualitzar les planificacions ja visualitzades a la plataforma. L'última fase és auto explicativa. Finalment, hi hauria una última fase de manteniment del sistema, que com no té un temps determinat no entra dins del TFG i no s'ha inclòs a la taula.

Fase	Tasques
Estudi previ	Estudi previ
Anàlisi i especificació de requisits	Anàlisi i especificació de requisits
<div>Construcció (comunicació Replan-Sistema-Plataforma)</div> <div>Construcció (comunicació Plataforma-Sistema-Replan)</div>	<div>[Per cada cas d'ús]</div> <div> Disseny ↓ Implementació ↓ Proves </div>
Elaboració de la memòria	Elaboració de la memòria

Taula 1. Metodologia de treball. (Font: elaboració pròpia)

4.2. Eines de seguiment

Com no es tracta d'un projecte a desenvolupar per un equip, sinó per una sola persona, no s'han tingut en compte eines de treball simultani, compartició de codi, etc.

Tanmateix, es treballarà amb eines de control de versions (Git), juntament amb gestors de repositoris en xarxa (GitHub), per tal de garantir la disponibilitat del codi i facilitar la recuperació en cas de fallades.

A més a més al final de cada dia de dedicació al projecte s'omplirà una fila en un full de càlcul a Google Drive on s'indicaran les tasques en les quals s'ha treballat aquell dia i la seva duració. D'aquesta manera es pot realitzar un seguiment de la feina feta que permetrà saber la velocitat en què avança el projecte i millorar l'organització dels futurs dies de dedicació al projecte.

Pel que fa a la comunicació amb els directors del projecte, s'utilitzaran eines habituals com l'e-mail per a la comunicació de missatges.

4.3. Mètode de validació

Les reunions regulars i freqüents amb els directors del projecte permetran anar revisant els requisits del projecte, tant la seva validesa com el seu compliment, tant funcionals com de qualitat. Aquestes reunions també garantirán una comunicació constant que facilitarà la resolució ràpida dels dubtes i problemes que puguin anar sorgint i alentint el ritme del projecte.

Es posarà a prova el *software* desenvolupat amb models de prova generats pels directors, de forma que es garanteix que es tindran en compte de forma objectiva els aspectes clau del funcionament del *software*.

5. Planificació temporal

5.1. Planificació general

5.1.1. Durada estimada del projecte

La durada estimada del projecte és de 4 mesos i mig aproximadament. El projecte va començar el 9 de gener de 2017 i es preveu la seva finalització pel 24 de maig del mateix any.

5.1.2. Recursos

Els recursos previstos per la realització d'aquest projecte són els següents:

Recursos personals: una persona, amb una dedicació al projecte de 25 hores setmanals aproximadament.

Recursos materials: un ordinador personal amb connexió a Internet i el *software* següent:

Recursos	Tipus	Finalitat
Google Docs Google Sheets Google Slides	Eina de desenvolupament	Per elaborar la documentació del projecte
Google Drive	Eina de desenvolupament	Per accedir a la documentació del projecte des de qualsevol dispositiu
Cacoo	Eina de desenvolupament	Per elaborar els diagrames i figures de la documentació
Adobe Photoshop CC 2014	Eina de desenvolupament	Per la creació i edició d'imatges
Correu electrònic Outlook	Eina de comunicació	Per mantenir comunicació amb els directors del TFG
Microsoft Project 2016	Eina de gestió	Per planificar el projecte
Acrobat Reader DC	Eina de desenvolupament	Per visualitzar documents
IntelliJ IDEA 2016 Ultimate Edition	Eina de desenvolupament	Per disposar d'un entorn de desenvolupament integrat que faciliti la implementació del projecte

Git	Eina de desenvolupament	Pel control de versions del repositori del codi font
Github	Eina de desenvolupament	Per disposar d'un repositori amb control de versions al núvol
Heroku	Eina de desenvolupament	Per desplegar l'aplicació web i el servei web i puguin ser utilitzats en un mode no local
Google Chrome	Eina de desenvolupament	Per testear l'aplicació web
Postman	Eina de desenvolupament	Per testear tant l'API creada com la que ofereix la plataforma de suport al desenvolupament col·laboratiu seleccionada

Taula 2. Recursos materials. (Font: elaboració pròpia)

5.1.3. Valoració d'alternatives i pla d'acció

Tant la data de lliurament del projecte com la planificació es preveuen flexibles i modificables segons el ritme de desenvolupament del projecte. Es deixarà un marge de contingència de 3 setmanes per solucionar imprevistos que hagin aparegut durant el transcurs del projecte i millorar la qualitat i completesa del *software* desenvolupat. En el cas que no es fes ús de tot el marge de contingència, es valorarà realitzar una fase addicional que permetés detectar canvis en els recursos d'una *release* en curs, com per exemple l'exclusió d'un recurs perquè ja no forma part de l'equip o està de baixa.

Les fases de construcció són les més susceptibles a patir desviacions temporals. L'origen d'aquestes desviacions pot ser la subestimació o sobreestimació de les hores dedicació assignades a la realització d'una tasca en el moment de realitzar la planificació del projecte. Es pot donar el cas que una tasca requereixi més temps de dedicació que la prevista a causa de la falta d'experiència del desenvolupador en les eines, llenguatges i *frameworks* que s'utilitzaran en el desenvolupament del projecte. També es pot donar el cas contrari, que en el moment de realitzar la planificació s'estimés que una tasca requereixi una dedicació superior però que en el moment de realitzar aquesta tasca s'acabi abans del previst.

Si les desviacions són de més de 5 dies, s'aplicaran mesures per resoldre aquesta situació i es realitzarà una nova planificació del projecte. Aquestes mesures consistiran a simplificar la funcionalitat que s'estigui implementant o en el cas de no ser possible cancel·lar la seva realització sempre que aquesta funcionalitat no sigui crítica. En el cas d'afectar una funcionalitat crítica s'endarreriria la implementació d'una funcionalitat no crítica que no afecti la qualitat del *software* a desenvolupar i, en el pitjor dels casos, es podria cancel·lar.

5.1.4. Consideracions globals

És important tenir en compte que aquest projecte serà realitzat per una única persona i que per tant no es podran realitzar tasques en paral·lel. Com a resultat de la metodologia emprada, la realització de les tasques serà de forma seqüencial en gran part del procés. Per tant, a part de les dependències particulars de cada tasca, serà necessari afegir a cadascuna d'elles la tasca anterior, ja que no es començarà una altra tasca fins que no acabi la prèvia, i es duran a terme en l'ordre en què apareixen en el següent apartat. Per tant, no s'ha trobat necessari fer un diagrama de Pert donat que el camí crític és l'únic camí existent i que el temps total del projecte és la suma d'hores de totes les tasques que es duran a terme.

5.2. Descripció de les tasques

5.2.1. Estudi previ

En aquesta primera fase inicial es contextualitzarà el projecte, el problema a resoldre i el propòsit del projecte. Aquesta fase no té cap dependència de precedència.

Les tasques a realitzar en aquesta fase són:

- Definició del projecte
- Estat de l'art
- Planificació temporal
- Gestió econòmica i sostenibilitat
- Introducció a l'entorn de treball

5.2.2. Anàlisi i especificació de requisits

En aquesta fase es definiran els requisits funcionals i de qualitat que ha de satisfer la solució *software*. Per tal de desenvolupar una solució que satisfaci aquests requisits, s'estudiarà tant la comunicació interna entre els diferents components dels sistemes com la comunicació entre sistemes i es proposaran possibles solucions que tinguin en compte tant les possibilitats que ofereixen les plataformes de suport al desenvolupament col·laboratiu escollides com les que ofereix el Replan. Una vegada s'han definit les possibles solucions, es valoraran i es triarà una tot justificant la seva elecció. Finalment, es definiran els casos d'ús de la solució escollida tenint en compte els diferents tipus d'usuaris. Aquesta fase té com a dependència de precedència la fase anterior.

Les tasques a realitzar en aquesta fase són:

- Definició de requisits
- Estudi de l'eina Replan
- Estudi de les plataformes
- Anàlisi de possibles solucions
- Definició casos d'ús

5.2.3. Construcció 1 (comunicació Replan-Sistema-Plataforma)

En aquesta fase es dissenyaran, implementaran i es testejaran els diversos casos d'ús que permeten visualitzar una planificació generada amb l'eina Replan a la plataforma escollida. D'aquesta manera no només el cap de projecte que ha generat la planificació és capaç de veure-la sinó que tot l'equip de desenvolupament en seria capaç. També és el primer pas per poder proveir al Replan amb l'estat de la feina que realitzen els desenvolupadors durant una *release* en marxa. Per poder visualitzar una planificació en qualsevol de les plataformes és necessari permetre als usuaris donar permís a l'aplicació per operar a través del seu compte i proporcionar a l'usuari una manera còmoda de relacionar els recursos que apareixen en les planificacions amb usuaris de la plataforma en qüestió.

Aquesta fase té com a dependència de precedència la fase anterior. Cadascun dels casos d'ús que s'implementaran en aquesta fase es realitzaran de forma seqüencial i per tant dependran de la finalització del cas d'ús anterior d'acord amb la metodologia emprada. Al seu torn, cada cas d'ús està dividit en 3 fases diferents (disseny, implementació i proves i avaluació) on cadascuna d'aquestes fases té com a dependència de precedència l'anterior.

Les tasques a realitzar en aquesta fase són cadascuna de les fases dels casos d'ús següents:

- Autorització de l'aplicació
- Visualització planificació a la plataforma

5.2.4. Construcció 2 (comunicació Plataforma-Sistema-Replan)

En aquesta fase es dissenyaran, implementaran i es testejaran l'enviament del *feedback* de l'activitat dels desenvolupadors a la plataforma escollida al Replan i l'actualització de les planificacions. També es desenvoluparan els casos d'ús que notifiquen als usuaris de tasques a punt d'expirar i retards en la finalització d'alguna d'elles.

Aquesta fase té com a dependència de precedència la fase anterior, s'aplica l'ordre seqüencial en implementar els casos d'ús i aquests estan dividits en les fases mencionades anteriorment. Les dependències de precedència s'apliquen de la mateixa forma que en la fase anterior.

Les tasques a realitzar en aquesta fase són cadascuna de les fases dels casos d'ús següents:

- Finalització tasca
- Rebuig tasca
- Actualització planificació
- Notificació tasca expirada no finalitzada
- Recordatori tasca a punt d'expirar

5.2.5. Elaboració de la memòria

En aquesta fase es posarà en ordre tota la documentació generada tant a la fase inicial, com a l'assignatura de GEP i durant el desenvolupament de les parts més tècniques del projecte amb l'objectiu d'elaborar el document de la memòria. En aquesta fase també es prepararà la defensa del projecte.

Les tasques a realitzar en aquesta fase són cadascuna de les fases dels casos d'ús següents:

- Elaboració de la memòria
- Preparació de la defensa

5.3. Desviacions en la planificació

Les fases d'**estudi previ**, **anàlisi i especificació de requisits** i part de la fase de **construcció 1** s'han ajustat a la planificació inicial. En la darrera de les fases mencionades hi ha hagut una desviació en la segona de les tasques a implementar (**visualització de la planificació a la plataforma**). Estava prevista la seva finalització el 24 de març i finalment ha estat finalitzada el 19 d'abril, el que significa que hi ha hagut un endarreriment de 18 dies.

Les causes d'aquesta desviació són les següents:

- En aquest cas d'ús l'usuari segueix un procés a la web en el qual ha d'introduir i seleccionar diferent informació per tal de poder visualitzar una planificació d'una release a Trello (que és la plataforma finalment escollida). En una part d'aquest procés, l'usuari ha de relacionar els recursos que participen en la planificació amb els usuaris de Trello corresponents a aquests recursos. En el moment d'elaborar la planificació inicial es va subestimar el temps necessari per realitzar aquesta part, tant en hores de disseny com d'implementació. Aquest fet va provocar que s'hagués de dedicar 2 dies més del previst.
- Durant realització d'aquesta tasca, el portàtil utilitzat per desenvolupar el projecte va patir una avaria i va ser necessari portar-lo a reparar. Aquesta circumstància va comportar la necessitat de preparar un ordinador de sobretaula que no disposava del software necessari per seguir amb el treball. Aquest fet només va causar una desviació d'1 dia respecte al previst en la planificació inicial.
- El desenvolupament d'aquesta tasca va coincidir amb nombroses entregues de la part final GEP que no s'havien tingut en compte en la planificació inicial. Mentre es treballava en aquests lliurables no es va avançar en la part tècnica del projecte, suposant un endarreriment de 5 dies.
- En una de les reunions de seguiment amb els directors que va tenir lloc durant realització d'aquesta tasca, es va veure que el disseny de la interfície no era el més adient per l'usuari i resultava fins i tot confús, ja que es trobaven tots els passos a realitzar en una mateixa pàgina. Es va concloure redissenyar el procés dividint-lo en diverses pantalles i mostrar més informació en alguns dels apartats, d'aquesta

manera la interfície web guanyaria en usabilitat sent més amigable per l'usuari. Aprofitant el redisseny de la interfície d'usuari, també es va redissenyar la persistència i es va millorar l'estructura del projecte. Es va trigar 7 dies a dur a terme aquest redisseny.

- La resta de dies desviació s'atribueixen a la falta d'experiència tant amb les tecnologies utilitzades, concretament AngularJS i els data repositories de Spring, com amb la maquetació del web.

5.4. Canvis en la planificació

Durant una reunió de seguiment amb els directors del projecte, va sorgir una nova funcionalitat que aporta molt de valor a l'usuari final. Aquesta funcionalitat, que ens referirem a ella com **Consultar canvis d'estat d'una tasca**, permetrà a l'usuari veure les etapes per les quals ha passat una tasca. D'aquesta manera l'usuari podrà saber si una tasca que ha estat marcada com finalitzada ha passat per totes les etapes o fins i tot veure si aquesta tasca ja havia estat rebutjada prèviament. Juntament amb la data en què s'ha canviat d'estat la tasca i altra informació com pot ser les persones que estaven realitzant-la, aquesta funcionalitat obre les portes a una possible explotació de les dades que pot ser d'ajuda per l'usuari, però que quedaria fora de l'abast d'aquest projecte.

S'ha optat per no implementar les tasques **Notificació tasca expirada no finalitzada** i **Recordatori tasca a punt d'expirar** per falta de temps. La primera d'elles s'ha optat per deixar-la fora perquè hi ha moltes casuístiques que fan que sigui necessària una replanificació, si bé és cert que en aquest projecte s'identifiquen, ens centrarem en els casos en què es marca com finalitzada una tasca, sigui abans de la data fixada o després de la data fixada, quedant la resta com a feina futura no inclosa dins d'aquest TFG. La segona d'elles, no aporta gaire valor, ja que el mateix Trello informa via email sobre aquesta circumstància.

També s'ha afegit una última fase que consistirà en l'elaboració de la memòria i la preparació de la defensa del projecte.

El retràs respecte a la planificació ha hagut de ser mitigat incrementant les hores diàries de dedicació del projecte de 5 a 7 per entregar la memòria dins del termini establert.

Els canvis en la planificació es poden veure en més detall a l'*Annex B*. En les figures d'aquest annex es pot veure com la durada del projecte s'ha vist incrementada un mes.

6. Gestió econòmica

6.1. Estimació de costos

En aquesta secció, es realitzarà una anàlisi dels recursos necessaris per a la realització del projecte i una estimació del cost total d'aquest. S'assumirà que és un projecte universitari, sense inversió econòmica i es tractarà el projecte com si fos un projecte competitiu, amb les hores dels recursos humans a preu de mercat.

6.1.1. Costos directes per activitat

Els costos directes per activitat en aquest projecte inclouen els recursos humans involucrats en les activitats definides en el diagrama de Gantt prèviament presentat. Encara que una sola persona desenvolupa el projecte encarregant-se de totes les activitats, cadascuna de les activitats correspon a un rol diferent i per tant les hores de cada rol tindran un preu de mercat diferent.

A la següent taula es poden veure els rols que intervenen en aquest projecte i el seu sou per hora calculat a partir d'ofertes laborals trobades a les pàgines Page Personnel^[11] i Michael Page^[12]:

Rol	Tipus de tasques	Sou [€/h]
Cap de projecte	Documentació i gestió	25
Analista	Anàlisi i disseny	15
Programador	Implementació i proves	12

Taula 3. Rols implicats en el desenvolupament del projecte. (Font: elaboració pròpia)

Cada tasca planificada la realitza un rol determinat i hi dedica unes hores determinades durant el desenvolupament del projecte.

Fase	Nom de la tasca	Hores estimades	Recurs	Cost [€]
Estudi previ	Definició del projecte	15	C	375
	Estat de l'art	20	C	500
	Planificació temporal	10	C	250
	Gestió econòmica i sostenibilitat	10	C	250
	Introducció a l'entorn de treball	20	C	500
	Total	75		1875
Anàlisi i especificació de requisits	Definició de requisits	10	A	150
	Estudi de l'eina Replan	15	A	225
	Estudi de les plataformes	20	A	300
	Anàlisi de possibles solucions	25	A	375
	Definició casos d'ús	15	A	225
	Total	85		1275
Construcció 1	Autorització de l'aplicació	30		390
	Disseny	10	A	150
	Implementació	15	P	180
	Proves	5	P	60
	Visualització planificació a la plataforma	85		1080
	Disseny	20	A	300
	Implementació	45	P	540
	Proves	20	P	240
	Total	115		1470
Construcció 2	Finalització tasca	40		510
	Disseny	10	A	150
	Implementació	20	P	240
	Proves	10	P	120
	Rebuig tasca	25		315
	Disseny	5	A	75
	Implementació	15	P	180
	Proves	5	P	60
	Actualització planificació	95		1185
	Disseny	15	A	225
	Implementació	60	P	720
	Proves	20	P	240
	Notificació tasca expirada no finalitzada	25		315
	Disseny	5	A	75
	Implementació	15	P	180
	Proves	5	P	60
	Recordatori tasca a punt d'expirar	30		375
	Disseny	5	A	75
	Implementació	15	P	180
	Proves	10	P	120
	Total	215		2700
Total del projecte		490		7320

Taula 4. Costos directes per activitat. (Font: elaboració pròpia)

6.1.2. Costos indirectes

- **Amortització del *hardware*:** l'únic *hardware* que s'utilitzarà és un ordinador portàtil que ja té 3 anys i que es considera amortitzat.
- **Transport:** s'utilitzarà el tren (Rodalies) i el metro amb abonaments de T-10 de dues zones^[13], que tenen un cost de 19,60 €. S'estima que s'anirà a la facultat una vegada cada 2 setmanes durant els 5 mesos que dura el projecte (11 viatges d'anada i 11 de tornada) per mantenir reunions amb els directors del projecte, 2 vegades per assistir a activitats de l'assignatura de GEP i 1 vegada per la defensa del projecte davant del tribunal. Per tant, es farien un total de 28 viatges sent necessàries 3 T-10 de dues zones.
- **Connexió a Internet:** s'utilitzarà connexió a Internet ja contractada i se suposarà que del total de l'amplada de banda consumit, un 25% es dedicarà al projecte.
- **Impressions en paper:** el lliurament del projecte inclou el lliurament en paper de la documentació d'aquest a cada un dels membres del tribunal. Se suposarà una extensió aproximada de 200 pàgines de memòria per a 3 membres del tribunal i els directors del projecte, a un cost de 0,05 € per pàgina (incloent-hi encuadernacions).

No hi ha costos associats al *software* utilitzat, ja que tot el que es farà servir serà lliure, gratuït o té una llicència gratuïta per estudiants de la FIB. No s'apliquen tampoc costos de local, subministraments d'aigua i llum, taxes, etcètera.

Producte	Unitats	Preu unitari	Percentatge de dedicació	Cost estimat
Transport	3 T-10	19,60 €	100	58,80 €
Connexió a Internet	4,5 mesos	29,75 €	25	33,47€
Impressions a paper	1000 pàgines	0,05 €	100	50€
Total				142,27€

Taula 5. Costos indirectes. (Font: elaboració pròpia)

6.1.3. Contingència

Es reservarà una part del pressupost per la partida de contingència, concretament un 15% de la suma dels costos directes i indirectes. Per tant el pressupost assignat a la partida de contingència és **1119,34 €**.

6.1.4. Imprevistos

- **Retard de 3 setmanes:** en el pla d'acció de la planificació s'ha mencionat deixar un marge de contingència de 3 setmanes per solucionar imprevistos. Suposem que el retràs implica hores d'analista (30%) i programador (70%) i hi ha una probabilitat del 30% que passi. És important recordar que la dedicació és de 25 hores setmanals.

- **Avaria de l'ordinador:** en cas que sorgís algun problema amb el *hardware*, s'haurà de reparar o substituir. Aquest contratemps pot incrementar el cost del projecte com a màxim el preu d'un ordinador nou (s'ha estimat oportú adquirir un portàtil nou valorat en 800 €). A aquest imprevist se li assignarà una probabilitat del 5%.

Imprevist	Probabilitat	Unitats	Cost
Retard de 3 setmanes	30	75 h	290,25 €
Avaria de l'ordinador	5	1	40€
Total			330,25 €

Taula 6. Costos imprevistos. (Font: elaboració pròpia)

6.1.5. Cost total

Cal mencionar que no es tenen en compte augments de preu (escalament) durant el transcurs del projecte, ja que no és de llarga duració. Tampoc s'afegeix cap marge de benefici sobre el cost total del projecte, ja que és un projecte sense ànim de lucre i no destinat a la venda de cap producte.

Concepte	Preu
Costos directes	7320€
Costos indirectes	142,27€
Contingència	1119,34€
Imprevistos	330,25€
Total	8911,86€

Taula 7. Cost total del projecte. (Font: elaboració pròpia)

6.2. Control de la gestió

Com ja s'ha comentat a la secció de metodologia i rigor, es mantindrà un registre de les hores de feina realitzades que recollirà per cada dia les tasques realitzades. D'aquesta manera es podran revisar les dades i comparar-les amb la planificació temporal per tal de veure si se sobrepassen les hores de dedicació i corregir possibles desviacions al llarg del projecte.

Al final del projecte es disposarà de dades definitives com la duració real del projecte, els recursos utilitzats i els imprevistos. Amb aquestes dades es farà un càlcul del cost real del projecte i es compararà amb els costos que s'han estimat en aquesta secció.

En el cas que el cost real del projecte sigui superior a l'estimat, es mirarà si la causa d'aquest augment del cost són els imprevistos tinguts en compte i si es pot cobrir aquesta diferència amb la partida corresponent. Altrament, s'assignarà la partida de contingència per tal de cobrir els costos ocasionats.

El càlcul de les desviacions es farà de la següent manera:

- Desviament en la realització d'una tasca en preu = (cost estimat – cost real) * consum d'hores real
- Desviament d'un recurs en preu = (cost estimat – cost real) * consum real
- Desviament en la realització d'una tasca en consum = (consum estimat – consum real) * cost real
- Desviament total en la realització de tasques = cost total estimat tasca – cost total real tasca
- Desviament total en recursos = cost total estimat recursos – cost total real recursos
- Desviament total costos fixos = cost total fix estimat – cost total fix real

Finalment s'utilitzarà la següent taula per calcular les desviacions per cadascuna de les fases definides:

Fase	Hores planificades	Hores reals	Δ Hores	Cost estimat [€]	Cost real [€]	Δ Cost [€]
Estudi previ	75	75	0	1875	1875	0
Anàlisi i especificació de requisits	85	85	0	1275	1275	0
Construcció 1	115	200	85	1470	2875	1405
Construcció 2	215	185	-30	2700	3165	465
Elaboració memòria	0	54	54	0	1350	1350
Total	490	599	109	7320	10540	3220

Taula 8. Repercussió de les desviacions en el cost final del projecte. (Font: elaboració pròpia)

Per tant, al pressupost inicial se li ha de sumar les desviacions que han anat apareixent al llarg del projecte.

També cal mencionar que es va avariar el portàtil sent necessària una reparació però aquesta reparació no repercuteix en el cost del projecte, ja que malgrat superar el cost assignat a l'imprevist, la partida de contingència ho cobreix.

Tenint en compte tot l'esmentat anteriorment, el cost final del projecte és **9377,68€**.

7. Elecció de la plataforma de suport

Per dur a terme el propòsit, cadascuna de les solucions requereixen:

- Una web simple a través de la qual es donarien permisos al plugin per integrar-la amb la plataforma corresponent.
- Un web service amb una API que permeti comunicar la plataforma de desenvolupament col·laboratiu amb el Replan.

A la següent figura es pot veure un esquema de l'arquitectura de les solucions:

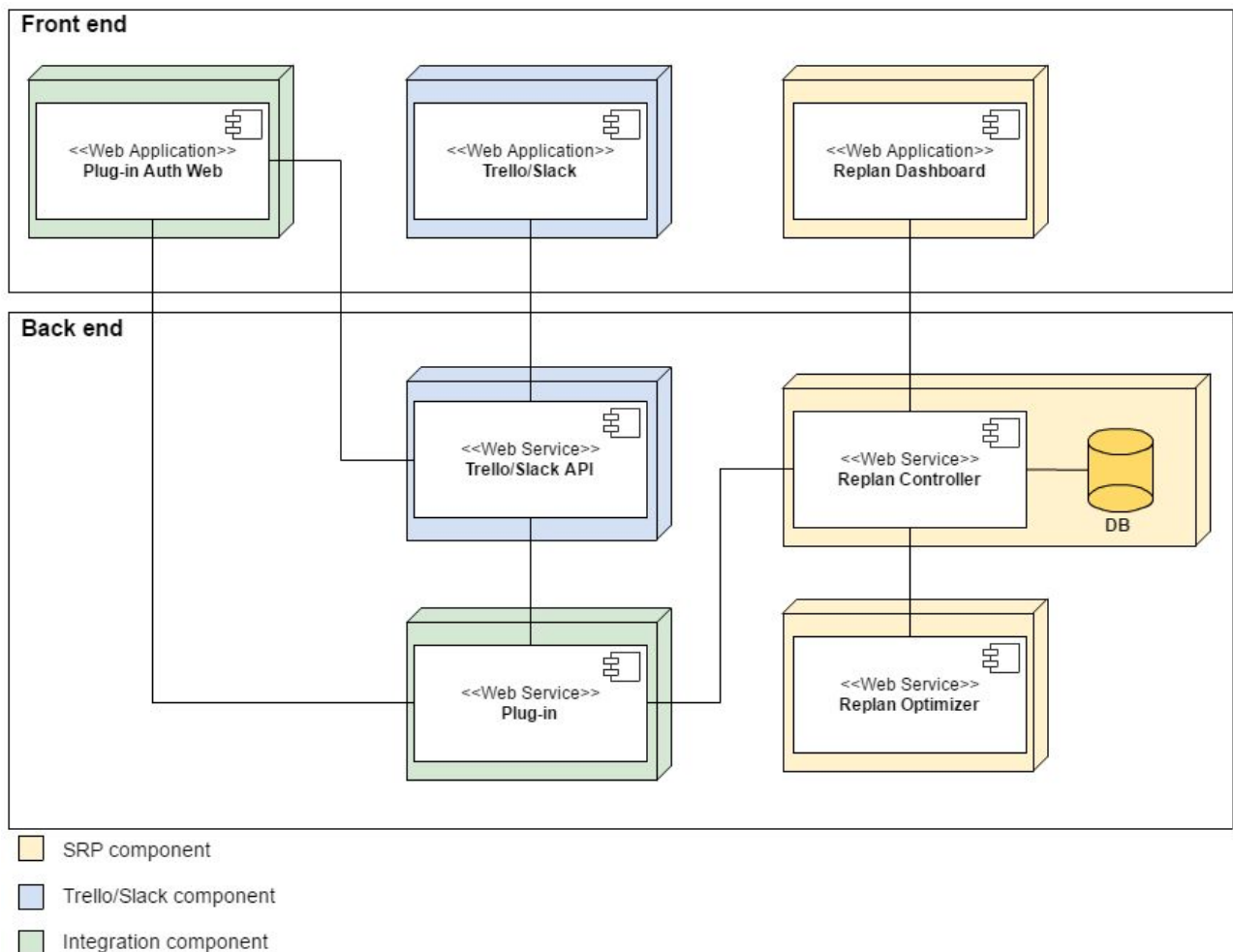


Fig 2. Arquitectura dels sistemes que intervenen (Font: elaboració pròpia).

7.1. Possibles solucions

7.1.1. Aplicació de Slack

La solució a Slack consisteix a crear una aplicació de Slack amb un bot integrat. La comunicació entre el Replan i els usuaris de Slack girarà entorn d'aquest bot. L'usuari podrà llistar les tasques provinents de la planificació generada pel Replan, assignar-les a canals i finalitzar-les escrivint una sèrie de comandes en el xat privat amb el bot. També es podrà convidar el bot als canals, allà els usuaris podran interaccionar amb ell per assignar tasques al canal, llistar-les i finalitzar-les.

A més a més, el bot també podrà informar els usuaris d'una nova generació de tasques per part del Replan a través del xat privat o per algun canal en concret.

7.1.1.1. Configuració inicial

Tot usuari de Slack que volgués utilitzar l'aplicació per primera vegada hauria de visitar la pàgina web d'instal·lació. L'usuari clica en un botó "Add to Slack" i seria redirigit a una pàgina oficial de Slack on hauria de fer login i donar permisos a l'aplicació. Finalment l'usuari seria redirigit a la pàgina on es trobava abans de l'autenticació.

Per tant, caldria disposar d'una web senzilla amb el botó "Add to slack". Aquest botó el proporciona la mateixa pàgina de l'API de Slack i cobriria una part del cicle d'autenticació. Una vegada l'usuari hagués realitzat correctament la fase d'autenticació i donat permisos, seria redirigit a la pàgina inicial. Per finalitzar el procés d'instal·lació, caldria que des de la part de client s'obtingués el "code" retornat a l'URL de redirecció s'enviés a la part de servidor per tal de fer el segon pas de l'autorització utilitzant el mètode oauth de l'API completant així el procés d'autorització.

7.1.1.2. Funcionament

Una vegada s'han donat permisos a l'aplicació, un bot apareixeria a la llista de contactes de l'equip. Ara l'usuari podria escriure una comanda en el xat privat amb el bot o en un canal on el bot estigui present per llistar tasques, assignar tasques a un canal o convidar el bot a un canal.

Llistar tasques és el cas més representatiu de l'ús de l'aplicació, ja que a partir d'aquesta acció es poden finalitzar les tasques. Cal recordar que el propòsit d'aquesta aplicació és informar el Replan quan una tasca ha estat finalitzada. Per aquest motiu a continuació es detallarà aquesta funcionalitat:

1. L'usuari escriu la comanda corresponent a llistar tasques en el xat privat amb el bot o en un canal on el bot està present.
2. Com s'havia definit prèviament un slash command a Slack per aquesta comanda, es realitzarà una petició HTTP POST a l'url definida a la slash command amb informació sobre la comanda i l'usuari que l'ha executat. L'url definida serà la corresponent a una de l'API del web service.
3. El web service realitzarà una crida a l'API del Replan i obtindrà les tasques de la planificació prèviament generada.
4. El web service retornarà les tasques a Slack a través d'un missatge del bot.
5. A la conversació o canal on s'hagi executat la comanda es visualitzarà un missatge del bot amb una llista de tasques. Cada tasca té un botó (interactive button) per marcar-la com a finalitzada i un altre per veure més informació.
6. L'usuari clica en el botó corresponent a la tasca que vol finalitzar.
7. Es comunica al Replan que la tasca ha estat finalitzada.

Un altre aspecte a destacar és que el bot no només serviria per establir una comunicació unidireccional entre els usuaris i el Replan sinó que també podria informar els usuaris de les noves planificacions realitzades pel Replan.

Com s'ha vist durant el transcurs de l'explicació del funcionament, l'aplicació utilitzaria moltes de les opcions que ofereix l'API de Slack: bot users, incoming webhooks, slash commands i interactive messages. Aquestes opcions s'habiliten i es configuren a l'app de Slack.

7.1.1.3. Valoració de la solució

La solució proposada no requereix una constant actualització de les tasques, ja que l'usuari les obté quan ho demana evitant així problemes de sincronicitat. També és important mencionar que l'usuari pot ser informat de la nova planificació generada pel Replan immediatament.

Malgrat els avantatges que ofereix aquesta solució i l'actual popularitat de Slack, Slack no és la plataforma més adient per realitzar la integració amb el Replan. Slack és una plataforma de comunicació a través de missatges entre membres d'un equip de treball i per tant és utilitzat amb aquesta finalitat. Afegir el concepte de tasca pot resultar d'utilitat, de fet hi ha aplicacions com *Busybot* que ja ho fan, però requereix una plataforma externa on visualitzar correctament la informació de les tasques. Un dels objectius d'aquest TFG és notificar al Replan de la finalització d'una tasca i creant una aplicació de Slack es pot aconseguir, com s'ha pogut veure en aquesta solució. Però els desenvolupadors i caps de projecte, que és a qui va dirigit aquest entorn Replan, no només utilitzen Slack (qui els utilitzin) sinó que també utilitzen plataformes de gestió i organització del treball com pot ser Trello que incorporen el concepte de tasca o similar.

7.1.2. Trello com a consumidor i notificador de canvis en l'estat de les tasques del Replan

Aquesta solució consisteix a utilitzar Trello com a plataforma consumidora de les planificacions generades pel Replan i també informaria de forma automàtica i transparent per l'usuari de la finalització d'una tasca. Per tant, caldrà utilitzar el Dashboard per poder generar les planificacions.

Per poder representar la informació obtinguda del Replan a Trello, el plugin a implementar estableix equivalències entre els conceptes bàsics dels dos entorns (Replan i Trello):

Concepte a Replan	Concepte a Trello
Release	Board
Feature	Card
Resource	Member

Taula 9. Equivalència de conceptes (Font: elaboració pròpia)

A Trello no es pot representar tota la informació que conté una planificació generada pel Replan però es pot representar la informació més rellevant. A la següent taula es pot veure com serà representada aquesta informació a Trello.

Informació	Representació a Trello
Membre assignat a la tasca	Usuari assignat a la tasca
Nom de la tasca	Títol de la card
Descripció de la tasca	Descripció de la card
Esforç	En el títol, davant del nom de la tasca i entre parèntesis
Data de venciment	Data de venciment de la card
Data d'inici	Descripció de la card, com a apartat
Tasques de les que depèn	Descripció de la card, com a apartat

Taula 10. Representació de la informació d'una tasca a Trello (Font: elaboració pròpia)

7.1.2.1. Configuració inicial

A continuació es detalla el procés d'inicialitzar el plugin:

1. Un usuari administrador del board de l'equip entra a la web d'autorització del plugin. El plugin utilitzarà l'usuari que li ha donat permís, per tant és important que sigui un usuari administrador, ja que així evitaria possibles restriccions.
2. Un cop a la web, omple el formulari de registre i prem el botó d'autoritzar.
3. Apareix un pop-up on l'usuari haurà d'autenticar-se i acceptar els permisos per utilitzar el plugin.
4. Si l'autenticació és correcta i l'usuari ha acceptat els permisos s'obté un token que es passarà a la part de servidor per tal que, juntament amb l'API key, el plugin pugui fer les crides a l'API de Trello. Gràcies a disposar d'aquest token, a la web es mostraran els boards de l'usuari.
5. L'usuari seleccionarà el projecte i la release de la qual vol carregar la planificació, l'equip de Trello, crearà associacions entre recursos i usuaris de Trello de l'equip seleccionat i introduirà el nom del nou board on vol que es mostrin les tasques del Replan i es monitorarà i finalment confirmarà.

7.1.2.2. Funcionament

Una vegada realitzada la configuració inicial, el plugin obtindrà les tasques del Replan i crearà les cards corresponents (juntament amb el seu webhook i llistes necessàries) que es mostraran en el board seleccionat per l'usuari.

Els webhooks assignats cada card fan que el monitoratge de les cards sigui possible, ja que cada vegada que es modifiqui una card es notificarà al web service. El web service detectarà si s'ha mogut una card de llista, que vol dir que ha canviat d'estat, i ho comunicarà al Replan en el cas que s'hagi mogut a la llista on van les cards finalitzades.

7.1.2.3. Valoració de la solució

Aquesta solució és còmode pels desenvolupadors, ja que no suposa canviar la seva manera d'utilitzar la plataforma perquè no han de realitzar accions extremes per informar el Replan de la finalització d'una tasca, la mouen a una llista concreta i és el plugin qui automàticament ho detecta perquè la finalització d'aquesta tasca sigui confirmada pel cap de projecte des de la web. A més a més, és possible actualitzar el tauler amb la replanificació procedent del Replan.

D'altra banda també té alguns aspectes que s'haurien de millorar. Un aspecte seria que per mostrar les releases a la web on es concedeix permís al plugin, perquè l'usuari la carregui en un board caldrà algun tipus d'associació entre l'usuari de Trello i la persona que ha generat la planificació de la release amb el Dashboard. També es pot solucionar fent que l'usuari introdueixi una url per tal de poder fer la comunicació amb el Replan.

Un altre aspecte és que no es pot informar els usuaris de noves planificacions, ja que l'API de Trello no permet crear notificacions, però sí que permetria actualitzar el board amb la nova planificació com s'ha comentat. Una forma de fer-ho seria creant una extensió pel navegador Chrome per veure aquestes notificacions, fins i tot es podria mostrar el nombre notificacions sense llegir. Si es creés aquesta extensió, podria substituir la web realitzant així la part de l'autorització del plugin a l'extensió. També existeixen les opcions de notificar als usuaris per mail o crear una llista en el tauler exclusivament per posar-hi cards de notificació.

És cert que Trello no permet crear notificacions, però sí que envia un correu als usuaris subscrits a una card quan resten 24 hores perquè aquesta expiri.

7.1.3. Trello com a interfície del Replan

La solució consistiria a utilitzar Trello com a plataforma d'entrada de les dades necessàries per generar una planificació. D'aquesta manera els usuaris podrien obtenir una planificació sense necessitat de sortir de la plataforma i el monitoratge de la finalització de les tasques, que funcionaria de la mateixa manera que la solució anterior.

La principal característica és que es necessita un component extern per tal que l'usuari pugui confirmar que ha acabat d'introduir la informació i vol obtenir una planificació. Per aquest motiu es necessitaria una extensió de navegador o una web senzilla que permetés a l'usuari fer tal cosa.

7.1.3.1. Configuració inicial

El procés d'inicialitzar el plugin és molt similar a la solució anterior, ja que és la manera en què s'autoritza una aplicació a Trello. On difereix és al 4t pas, s'obtindrà el token però el que vindria a continuació ja no forma part de la configuració inicial.

7.1.3.2. Funcionament

Després que l'usuari ha donat permisos al plugin, l'usuari necessita disposar d'un board amb el qual es farà la planificació. Si no en tenia cap de preparat, en aquest moment crearia un de nou i l'ompliria amb les cards que consideri oportunes.

A continuació, l'usuari visitarà la web o obrirà l'extensió de navegador on triarà el board del qual vol una planificació, introduirà la informació d'una release (nom, descripció i deadline) i premerà un botó per confirmar. En conseqüència, el plugin a implementar rebrà una petició de generació de planificació. En el cas que les cards continguessin la informació necessària introduïda de forma correcta, el plugin crearà webhooks per cadascuna de les cards del board per monitorar l'estat d'aquestes i enviarà la informació al Replan perquè generi la planificació per aquelles dades. Si aquest no fos el cas, el plugin informaria dels errors o de la informació mancanta a través de missatges a l'extensió de navegador.

Una vegada el plugin obté la planificació del Replan, assignarà les cards als membres segons el que especifica la planificació.

7.1.3.3. Valoració de la solució

No haver d'utilitzar una altra plataforma diferent per definir la planificació com seria el dashboard resulta molt atractiu per l'usuari però aquesta solució té una feblesa: la introducció de les dades.

La informació que necessita el Replan per generar una planificació és difícil d'introduir utilitzant Trello. Per algunes dades com els punts d'història, les habilitats necessàries o les tasques de les quals depèn una tasca, Trello no ofereix una manera de representar-ho i per tant per especificar aquesta informació caldria escriure d'una manera molt concreta a la descripció de les cards per exemple. Això comportaria que els usuaris cometessin errors i per tant caldria detectar aquests errors en les cards del board seleccionat i informar a l'usuari de l'error en concret perquè l'arregli, avortant la generació de la planificació. Aquest fet podria provocar que l'usuari sentís que està perdent el temps.

Un altre aspecte a considerar és que la selecció dels recursos per una release no es podria fer a Trello i s'hauria de fer a l'extensió o web.

Finalment, igual que a la solució anterior, no es podria alertar a través de Trello de noves planificacions.

7.2. Justificació de la solució escollida

A partir del que s'ha comentat a les valoracions i de cada solució i tenint en compte el temps disponible la millor solució seria ***Trello com a consumidor i notificador de canvis en l'estat de les tasques del Replan***. El que ha fet decantar-me per aquesta opció és que els usuaris no han de canviar la seva forma de treballar amb Trello i no es veuen involucrats en el procés de notificar de la finalització d'una tasca sent així una solució usable. Un altre factor és la sincronització del tauler de Trello entre tots els usuaris i amb el plugin que es desenvoluparà. Finalment, el no poder enviar notificacions de noves planificacions generades pel Replan utilitzant les notificacions de Trello tampoc suposa un greu problema perquè es podria solucionar enviant una notificació en forma de correu per exemple. Cal recordar que independentment de notificar als usuaris o no, es pot actualitzar el tauler amb la planificació.

8. Representació de la planificació a Trello

Com ja s'ha comentat al principi d'aquest document, Replan genera planificacions d'una release d'un projecte de *software*. Aquestes planificacions permeten saber quines tasques s'han d'implementar durant la release, quan està previst que es comencin i s'acabin, de la finalització de quina altra tasca depenen (si és el cas) i quins són els recursos humans que tenen assignats.

Tenint en compte la informació que contenen les planificacions, l'estudi que s'ha realitzat sobre Trello i els objectius que vol assolir aquest projecte, s'ha pensat una manera de representar les planificacions a Trello aprofitant les seves característiques. És important mencionar que no tota la informació d'una planificació serà representada, només aquella que pugui ser d'utilitat als desenvolupadors.

8.1. Elements de Trello

8.1.1. Tauler

A Trello, un tauler o board és on s'organitzen els projectes, es comparteix informació i es treballa. Gràcies als taulers, tots els membres que pertanyin al tauler tenen una visió compartida de la feina que s'està realitzant.

Aprofitant aquestes característiques, les planificacions seran “carregades” en els taulers mitjançant el plugin. El plugin crearà un nou tauler amb llistes per defecte que contindran les targetes corresponents a les tasques de la planificació. Gràcies a això, els desenvolupadors podran visualitzar la planificació, veure l'estat en què es troba la release actualment i fer el seguiment de la seva feina. Per tant, el disseny del tauler esdevé crucial.

A les captures següents es pot veure una planificació carregada a Trello mitjançant el plugin en el moment inicial:

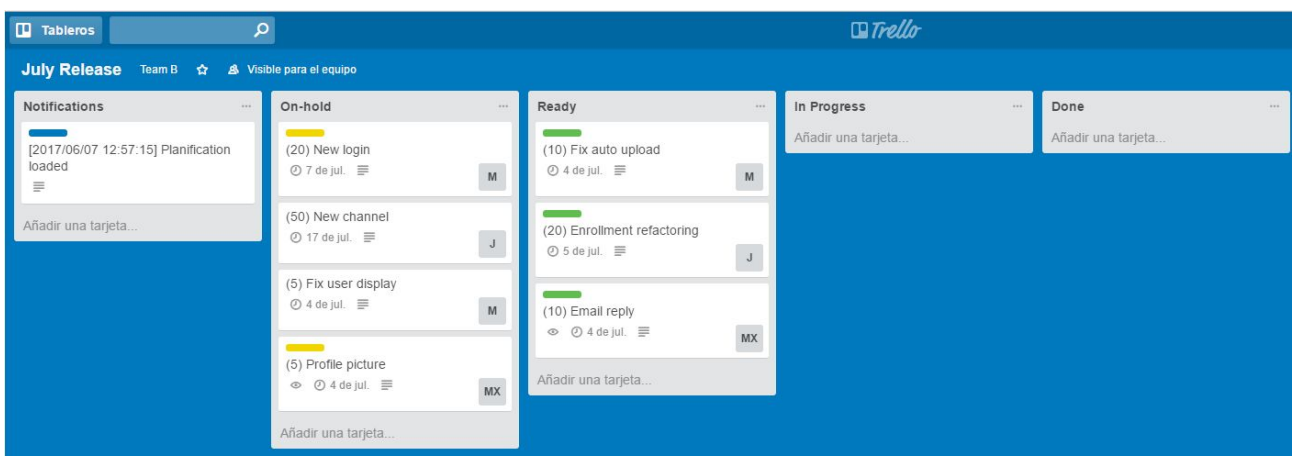


Fig 3. Planificació d'una release a Trello (moment inicial) (Font: elaboració pròpia)

A la part superior esquerra es pot veure el nom del tauler (“July Release”, en aquest cas) que és donat pel cap de projecte durant el procés de carregar la planificació a Trello. Com s’havia mencionat abans, el tauler conté llistes i targetes però s’entrarà en més detalla en les seccions posteriors.

8.1.2. Targeta

Les targetes o cards són elements a Trello que sovint representen tasques. En el nostre cas hi ha dos tipus de targetes: les targetes que representen una tasca de la planificació i les targetes informatives que tenen com a objectiu informar els usuaris d’esdeveniments que no són propis de Trello que més endavant es detallaran.

Les targetes que representen tasques contenen la informació extreta de la planificació de la següent forma:

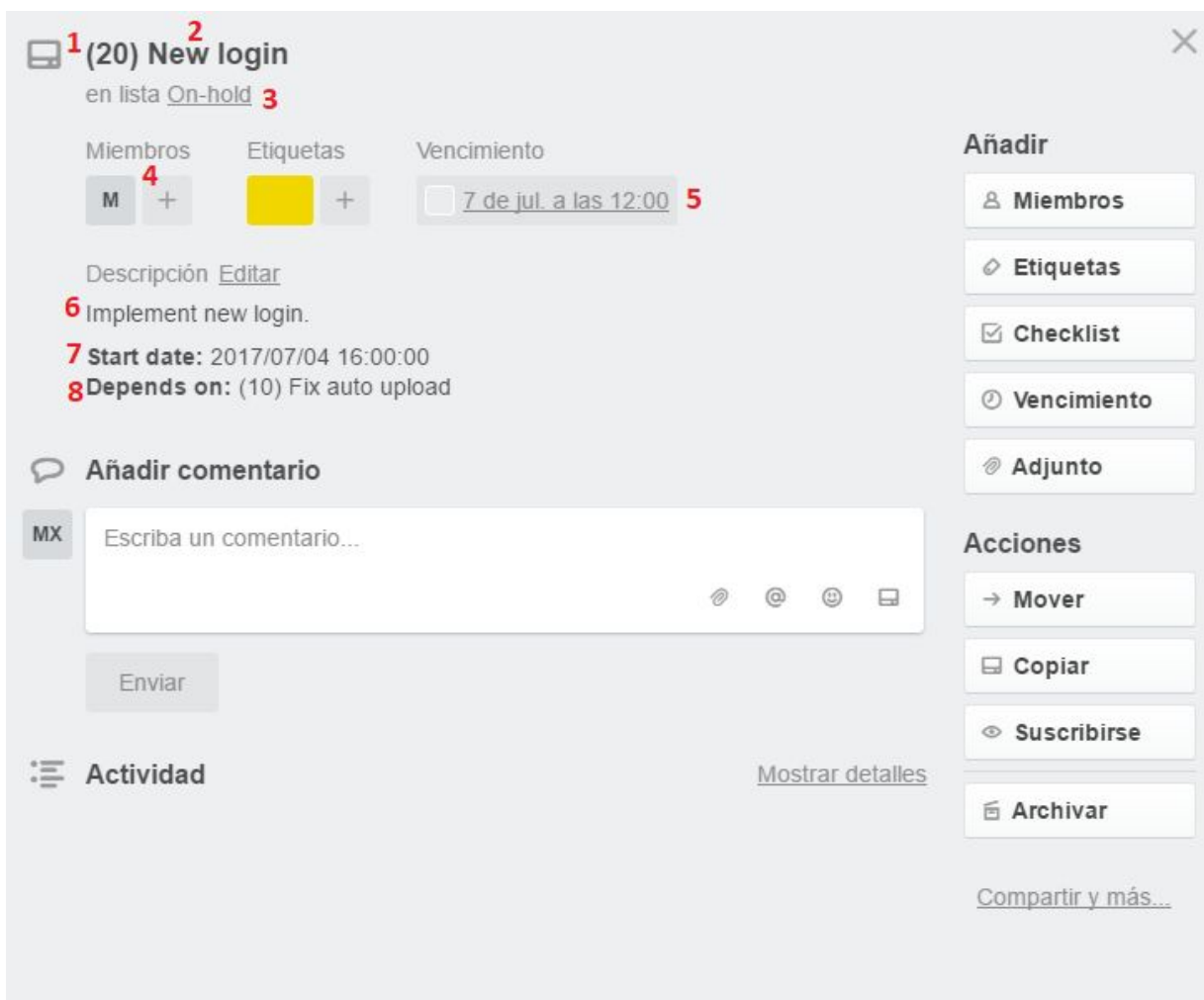


Fig.4 Detall d’una targeta de Trello. (Font: elaboració pròpia)

A la següent taula es detalla la informació que conté una targeta i la seva representació:

#	Informació	Representació a la targeta
1	Esforç	En el títol, davant del nom de la tasca
2	Nom de la tasca	Títol
3	Estat actual	Llista on es troba
4	Recursos assignats a la tasca	Usuaris assignats
5	Data de venciment	Data de venciment
6	Descripció de la tasca	Descripció
7	Data d'inici	Com a apartat <i>Start date</i> dins de la descripció
8	Tasques de les quals depèn	Com a apartat <i>Depends on</i> dins de la descripció

Taula 11. Representació de la informació d'una tasca en una targeta de Trello (Font: elaboració pròpia)

A part de la informació de les tasques, a les targetes s'utilitzen etiquetes de diferents colors per facilitar la comprensió del tauler:

- Groc: depèn d'una altra targeta
- Verd: llesta per ser implementada
- Lila: finalitzada pendent de validació
- Vermell: rebutjada la seva finalització

Amb les targetes s'utilitza el concepte de **next card** d'un desenvolupador. Aquest concepte s'ha creat per evitar que un desenvolupador estigui treballant en més d'una targeta alhora. La next card d'un desenvolupador és una targeta:

- No finalitzada
- Que actualment no depèn de la finalització d'una altra
- La que té la data més baixa de les targetes que té assignades un desenvolupador

8.1.3. Llista

Com s'ha comentat, quan es carrega una planificació a Trello, el plugin crea unes llistes per defecte. El propòsit d'aquestes llistes és oferir als usuaris una visualització aclaridora de la planificació i afavorir l'organització de les tasques a realitzar. Als usuaris familiaritzats amb Trello no els hi suposarà un gran canvi, ja que a Trello les llistes s'utilitzen per representar un procés o flux de treball i les targetes es mouen entre elles a mesura que avancen. En el nostre cas, les llistes representen estats pels quals va passant una targeta fins que és completada.

A l'hora de decidir el nombre de llistes per defecte i quines serien, s'han tingut en compte les necessitats dels desenvolupadors i reduir el nombre de llistes per evitar confondre als usuaris. Les llistes que es crearan per defecte són les següents:

- **Notifications:** conté targetes que informen els usuaris d'esdeveniments importants relacionats amb l'ús del plugin i que, per tant, l'historial d'activitat de Trello no és capaç de reflectir. Aquests esdeveniments són els següents:
 - Creació del tauler.
 - Actualitzacions del tauler a causa de replanificacions.
 - Targetes que han estat rebutjades com acabades.

A més a més, per fer-ho més visual, cada tipus de notificació tindrà una etiqueta d'un color diferent.

- **On-Hold:** conté les targetes que:
 - Depenen de la finalització d'una altra per ser realitzades.
 - No depenen de la finalització d'una altra per ser realitzades però no són la **next card** d'un dels desenvolupadors.

Si una targeta compleix el segon cas, aquesta targeta es mourà automàticament a la llista Ready sempre que sigui la **next card** d'un dels desenvolupadors.

- **Ready:** conté les targetes que no depenen de la finalització de cap altra per ser realitzades, són la **next card** d'un dels desenvolupadors i que no han estat començades. És a dir, les targetes que ja es poden començar a implementar.
- **In Progress:** conté les targetes en les quals l'equip està treballant actualment.
- **Done:** conté les targetes finalitzades per l'equip tècnic.

Per tal d'evitar un ús incorrecte del tauler per part dels membres de l'equip, s'establiran diverses limitacions en el moviment de targetes entre llistes. A continuació es detallen els moviments considerats vàlids:

- De Ready a In Progress
- De In Progress a Done

8.2. Replanificació

Recordant apartats anteriors, un dels objectius d'aquest projecte és proporcionar al Replan la informació sobre aquelles activitats que causen que sigui necessari una actualització (replanificació) de la release en curs. Per tant, el plugin no només ha de permetre al cap de projecte decidir en quin moment replanificar sinó que també ha de ser capaç de reflectir els canvis produïts per la replanificació en el tauler corresponent.

Per tal d'entendre millor les replanificacions, cal tenir present quines són les activitats que realitzen els desenvolupadors a Trello que causen que sigui necessària una replanificació. Aquestes activitats són:

- Una tasca és finalitzada abans de la data esperada: en aquest cas seria convenient una replanificació perquè potser es poden avançar tant la data de començament com la de finalització de les tasques, fet que permetria incorporar noves tasques a la releases en curs.

- Una tasca és finalitzada després de la data esperada: aquest cas significaria que s'ha produït un retard en la implementació d'aquesta tasca, fet que faria necessària una replanificació per tal de retardar les dates de començament i finalització d'algunes tasques, fer canvis en les assignacions de recursos a tasques i fins i tot descartar la implementació de tasques que estaven assignades a la release en curs.

Una vegada vists aquests casos, podem veure que les replanificacions poden comportar:

- Canvis en les dates de començament i finalització de les targetes.
- Canvis en l'assignació de membres a les targetes.
- L'aparició de noves targetes que no existien abans de la replanificació.
- La desaparició de targetes que sí que existien abans de la replanificació.
- Canvis de llista de les targetes perquè s'ha modificat la next card d'un membre, per tant targetes que estaven a la llista Ready poden passar a la llista On-hold i a l'inrevés.

9. Especificació del sistema

9.1. Requisits funcionals

En aquesta secció s'especificaran els casos d'ús que guarden una relació directa amb el plugin per Trello que s'implementarà. Es distingiran els casos d'ús segons el sistema al qual corresponguin, més concretament l'aplicació web utilitzada pel cap de projecte per gestionar les planificacions en curs i la plataforma de suport al desenvolupament col·laboratiu Trello on els desenvolupadors realitzen el seguiment del seu treball.

Els actors que participen en els casos d'ús són els següents:

- **Cap de projecte:** encarregat/da de la planificació de les *releases* d'un projecte i de revisar si les tasques implementades compleixen els requisits definits.
- **Desenvolupador:** membres de l'equip que realitzen les tasques tècniques.

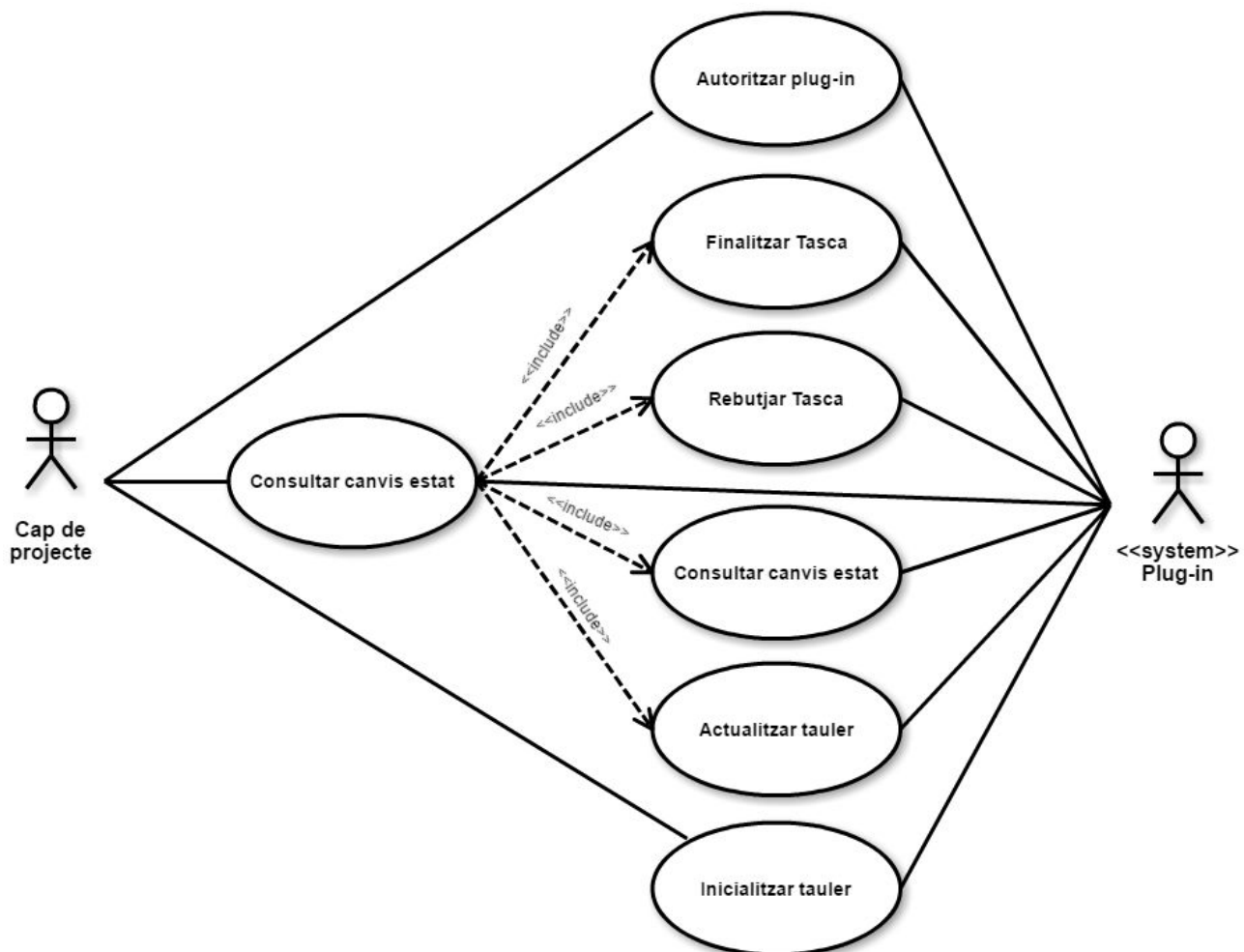


Fig 5. Diagrama de casos d'ús aplicació web (Font: elaboració pròpia)

9.1.1. Casos d'ús aplicació web

9.1.1.1. Autoritzar plugin

Actor principal: Cap de projecte

Precondicions: El cap de projecte disposa d'un compte a Trello però no disposa d'un compte a l'aplicació web.

Disparador: El cap de projecte vol integrar l'eina Replan a Trello.

Escenari principal:

1. El cap de projecte accedeix a la pàgina de registre.
2. La web mostra un formulari amb dos camps de text on es demana introduir un usuari i una contrasenya i un botó que permet autoritzar el plugin utilitzar el seu compte de Trello.
3. El cap de projecte omple els camps i clica el botó d'autoritzar.
4. S'obre una finestra emergent que informa dels permisos dels quals disposarà el plugin en el cas de la seva autorització i que conté dos botons: un botó per iniciar sessió a Trello i continuar amb el procés d'autorització i un altre per no permetre-ho.
5. El cap de projecte clica el botó d'iniciar sessió.
6. El cap de projecte és redirigit a una pàgina de log-in de Trello.
7. El cap de projecte introdueix el correu o usuari de Trello i la contrasenya del seu compte de Trello i confirma.
8. Trello valida les dades introduïdes pel cap de projecte
9. Es mostra una web amb la mateixa informació sobre els permisos que tindrà l'aplicació un cop autoritzada i dos botons: un botó que permet autoritzar al plugin utilitzar el seu compte i un altre per no permetre-ho.
10. El cap de projecte clica el botó de permetre.
11. Es tanca la finestra emergent i el plugin està autoritzat a utilitzar el compte del cap de projecte.
12. El cap de projecte clica el botó "Register" confirmant el registre.
13. El plugin valida les dades.
14. El cap de projecte accedeix a la web amb les seves credencials.

Extensions:

- 4a. El cap de projecte té una sessió iniciada a Trello.
 - 4a1. Es salta al punt 9.
- 5a. El cap de projecte clica el botó de no permetre.
 - 5a1. Es torna al punt 3.
- 8a. Trello no valida les dades introduïdes pel cap de projecte.
 - 8a1. La web de log-in de Trello informa que les dades introduïdes no són correctes.
 - 8a2. Es torna al punt 5.
- 10a. El cap de projecte clica el botó de no permetre.
 - 10a1. Es torna al punt 3.
- 13a. El plugin no valida les dades.
 - 13a1. A la web es mostra un avís informant que ja existeix un usuari amb el nom introduït.

9.1.1.2. Inicialitzar tauler

Actor principal: Cap de projecte

Precondició: El cap de projecte ha iniciat sessió a la web.

Disparador: El cap de projecte vol carregar una planificació d'una *release* a Trello.

Escenari principal:

1. El cap de projecte clica sobre l'enllaç "Load to Trello" situat en el menú de navegació.
2. La web mostra un formulari amb tres desplegable amb opcions per defecte escollides que permeten escollir l'endpoint, el projecte i la release de la qual es vol carregar la planificació a Trello.
3. El cap de projecte selecciona l'endpoint que vol utilitzar.
4. La web mostra els projectes de l'endpoint seleccionat en les opcions del desplegable d'escollir projecte en el qual hi ha un projecte seleccionat per defecte. També mostra les releases del projecte seleccionat per defecte en les opcions del desplegable d'escollir release.
5. El cap de projecte selecciona un projecte de l'endpoint prèviament seleccionat.
6. La web mostra les releases del projecte seleccionat en les opcions del desplegable d'escollir release en el qual hi ha una release seleccionada per defecte.
7. El cap de projecte selecciona una release del projecte prèviament seleccionat.
8. El cap de projecte clica el botó "Next" per continuar el procés de carregar la planificació a Trello.
9. La web mostra un formulari amb un desplegable amb els equips de Trello als quals pertany el cap de projecte amb un equip seleccionat per defecte i una llista amb els membres de l'equip seleccionat.
10. El cap de projecte selecciona l'equip de Trello els membres del qual formen part de la release.
11. La web mostra la llista amb els membres de l'equip seleccionat.
12. El cap de projecte clica el botó "Next" per continuar el procés de carregar la planificació a Trello.
13. La web mostra una llista amb els "matchings" (relacions entre recursos involucrats en la planificació i els seus usuaris a Trello) enregistrats en el sistema pel cap de projecte (si n'hi ha), una taula amb els recursos que prenen part en la planificació de la release seleccionada prèviament però que no estan vinculats a cap usuari de Trello de l'equip seleccionat prèviament i una altra taula amb els usuaris de Trello de l'equip que no estan vinculats a cap recurs.
14. El cap de projecte selecciona un recurs de la taula i l'usuari de Trello corresponent de l'altra taula i clica el botó "Match".
15. S'emmagatzema la relació en el sistema, la web mostra la llista de matchings actualitzada i el recurs i l'usuari seleccionats desapareixen de les taules corresponents.
16. El cap de projecte clica el botó "Next" per continuar el procés de carregar la planificació a Trello.
17. La web mostra un formulari amb un camp de text per introduir el nom que se li vol donar al tauler que es crearà a Trello, el qual representarà la planificació a Trello.

18. El cap de projecte clica el botó "Load to Trello".
19. El plugin crea un tauler a Trello amb el nom introduït. Aquest tauler conté les llistes predefinides mencionades anteriorment en el document i les cards corresponents a les tasques de la planificació amb la informació comentada també anteriorment i en les llistes corresponents. A més a més, assigna als usuaris que formen part de l'equip seleccionat al tauler i a les cards corresponents segons la planificació. També crea una card notificació a la llista *Notifications* que informa de la inicialització del tauler.
20. La web mostra el nom de la release la planificació de la qual s'ha carregat a Trello i un enllaç per accedir al tauler.
21. El cap de projecte accedeix al tauler creat a través de l'enllaç.
 - *El punt 21 és opcional
 - *En qualsevol pas entre el 10 i el 18 (ambdós inclosos) es pot tornar al pas anterior del procés si es clica el botó "Previous".
 - *Els passos 14 i 15 es poden repetir diverses vegades.

Extensions:

- 2a. No s'han trobat endpoints.
 - 2a1. La web mostra els 3 desplegable i cada desplegable informa que no s'han trobat endpoints, projectes ni releases respectivament.
 - 2b1. S'acaba el cas d'ús.
- 2b. L'endpoint seleccionat no té projectes.
 - 2b1. La web mostra els 3 desplegable: el desplegable d'endpoints té una opció seleccionada per defecte mentre que els altres 2 desplegable informen que no s'han trobat projectes ni releases respectivament.
 - 2b2. S'acaba el cas d'ús.
- 2c. El projecte seleccionat no té releases.
 - 2c1. La web mostra els 3 desplegable: el desplegable d'endpoints i el desplegable de projecte tenen una opció seleccionada per defecte mentre que el desplegable de releases informa que no s'han trobat releases per aquest projecte.
 - 2c2. S'acaba el cas d'ús.
- 3a. El cap de projecte no selecciona cap endpoint perquè l'endpoint seleccionat per defecte és el que vol.
 - 3a1. Es salta al punt 5.
- 4a. L'endpoint seleccionat no té projectes.
 - 4a1. La web mostra els 3 desplegable: el desplegable d'endpoints té una opció seleccionada per defecte mentre que els altres 2 desplegable informen que no s'han trobat projectes ni releases respectivament.
- 4b. El projecte seleccionat no té releases.
 - 4b1. La web mostra els 3 desplegable: el desplegable d'endpoints i el desplegable de projecte tenen una opció seleccionada per defecte mentre que el desplegable de releases informa que no s'han trobat releases per aquest projecte.
- 5a. El cap de projecte no selecciona cap projecte perquè el projecte seleccionat per defecte és el que vol.
 - 5a1. Es salta al punt 7.
- 6a. El projecte seleccionat no té releases.

6a1. La web mostra els 3 desplegable: el desplegable d'endpoints i el desplegable de projecte tenen una opció seleccionada per defecte mentre que el desplegable de releases informa que no s'han trobat releases per aquest projecte.

7a. El cap de projecte no selecciona cap release perquè la release seleccionada per defecte és la volguda.

8a. El cap de projecte no vol carregar la release seleccionada a Trello.

8a1. Es torna al punt 3.

9a. El cap de projecte no pertany a cap equip de Trello.

9a1. La web informa que l'usuari no pertany a cap equip de Trello

10a. El cap de projecte no selecciona cap equip de Trello perquè el seleccionat per defectes és el que vol.

10a1. Es salta al punt 12.

14a. El cap de projecte vol desvincular un recurs d'un usuari de Trello.

14a1. El cap de projecte selecciona el matching de la llista.

14a2. La web mostra el matching seleccionat amb un fons vermell.

14a3. El cap de projecte clica el botó "Unmatch".

14a4. El sistema engistra que s'ha desfet la relació entre el recurs i l'usuari de Trello, desapareix aquest matching de la llista i apareixen el recurs i l'usuari de Trello a les taules corresponents.

14a5. Es torna al punt 13.

14b. El cap de projecte considera adients els matchings actuals.

14b1. Es salta al punt 16.

17a. El cap de projecte ha clicat el botó "Next" sense que tots els recursos de la planificació estiguin relacionats amb un usuari de Trello.

17a1. La web mostra un quadre de diàleg informant que hi ha recursos que no han estat relacionats amb un usuari de Trello i si es desitja continuar.

17a2. El cap de projecte clica "Cancel·lar"

17a3. Es torna al punt 15.

17a2a. El cap de projecte clica "Acceptar"

17a2b. Es salta al punt 17.

9.1.1.3. Consultar canvis estat

Actor principal: Cap de projecte

Precondicions: El cap de projecte ha iniciat sessió a la web i es troba a la pàgina home.

Disparador: El cap de projecte vol integrar l'eina Replan a Trello.

Escenari principal:

1. La web mostra els moviments de cards entre llistes (canvis d'estat) que han realitzat els desenvolupadors en tots els taulars creats pel cap de projecte i les accions d'aquells moviments que correspongui.
2. El cap de projecte vol deixar de veure els canvis d'estat.

Extensions:

- 1a. El cap de projecte no ha creat cap taular a partir de la web.
 - 1a1. La web no mostra cap moviment de cards.
 - 1a2. S'acaba el cas d'ús.
- 1b, 2b2a Cap desenvolupador ha mogut una card d'una llista a una altra.
 - 1b1. La web no mostra cap moviment de cards.
 - 1b2. El cap de projecte prem el botó de "Refrescar"
 - 1b3. Es torna al punt 1.
- 2a. El cap de projecte selecciona la pestanya "Pending Actions".
 - 2a1. La web mostra els moviments de cards que requereixen una acció del cap de projecte, és a dir, els moviments a la llista *Done*.
 - 2a2. El cap de projecte vol rebutjar una tasca perquè considera que no ha estat finalitzada.
 - 2a2a. Rebutjar tasca.
 - 2a2b. Es torna al punt 1.
 - 2a3. El cap de projecte vol acceptar una tasca com finalitzada.
 - 2a3a. Finalitzar tasca.
 - 2a3b. Es torna al punt 1.
- 2b. El cap de projecte vol veure els canvis d'estat d'un taular en concret.
 - 2b1. El cap de projecte selecciona un taular.
 - 2b2. La web mostra els moviments de cards del taular seleccionat.
- 2c. El cap de projecte vol veure els canvis d'estat d'una tasca en concret.
 - 2c1. Consultar canvis estat tasca.
 - 2c2. Es torna al punt 1.
- 2d. El cap de projecte vol actualitzar la planificació actual i que es vegi reflectit a Trello.
 - 2d1. Actualitzar planificació.
 - 2d2. Es torna al punt 1.

9.1.1.4. Finalitzar tasca

Actor principal: Cap de projecte

Precondicions: El cap de projecte ha iniciat sessió a la web, es troba a la pàgina home i hi ha hagut moviment de cards entre llistes per part dels desenvolupadors.

Disparador: El cap de projecte vol donar per finalitzada una tasca o deixar de donar per finalitzada una tasca.

Escenari principal:

1. El cap de projecte clica el botó d'acceptar tasca com a finalitzada associat al moviment d'una card a la llista *Done* que no està marcat com finalitzat.
2. El plugin enregistra que s'ha acceptat la tasca corresponent al registre del moviment de la card.

Extensions:

1a. El cap de projecte vol deixar de donar per finalitzada una tasca.

1a1. El cap de projecte clica el botó d'acceptar tasca com a finalitzada associat al moviment d'una card a la llista *Done* que està marcat com finalitzat.

1a2. El plugin enregistra que ja no es considera com acceptada la tasca corresponent al registre del moviment de la card.

9.1.1.5. Rebutjar tasca

Actor principal: Cap de projecte

Precondicions: El cap de projecte ha iniciat sessió a la web, es troba a la pàgina home, hi ha hagut moviment de cards entre llistes per part dels desenvolupadors.

Disparador: El cap de projecte vol rebutjar una tasca perquè considera que no compleix els requisits necessaris per considerar-la finalitzada.

Escenari principal:

1. El cap de projecte clica el botó de rebutjar tasca associat al moviment d'una card a la llista *Done* que no està marcat com finalitzat.
2. La web mostra un formulari amb una àrea de text.
3. El cap de projecte omple l'àrea de text amb la seva justificació sobre el rebuig d'aquesta tasca i confirma que vol rebutjar la tasca i crear el comentari.
4. El plugin enregistra que s'ha rebutjat la tasca corresponent al registre del moviment de la card i per tant que no té accions pendents associades, mou la card rebutjada a la llista *In Progress*, substitueix l'etiqueta lila per una vermella i crea un comentari a la card amb el contingut que ha escrit el cap de projecte.

Extensions:

3a. El cap de projecte s'adona que no vol rebutjar la tasca.

3a1. El cap de projecte surt del formulari.

3a2. S'acaba el cas d'ús.

9.1.1.6. Consultar canvis estat tasca

Actor principal: Cap de projecte

Precondicions: El cap de projecte ha iniciat sessió a la web, es troba a la pàgina home, hi hagut moviment de cards entre llistes per part dels desenvolupadors.

Disparador: El cap de projecte vol consultar els estats pels quals ha passat una tasca.

Escenari principal:

1. El cap de projecte clica el botó de més informació associat al moviment d'una card.
2. La web mostra una taula on per cada fila es pot veure l'usuari que ha mogut la card, la llista a la qual l'ha mogut i la data.

Extensions:

-

9.1.1.7. Actualitzar tauler

Actor principal: Cap de projecte

Precondicions: El cap de projecte ha iniciat sessió a la web, es troba a la pàgina home, hi ha hagut moviment de cards entre llistes per part dels desenvolupadors, ha seleccionat un tauler i com a mínim una tasca corresponent a aquest tauler es troba acceptada.

Disparador: El cap de projecte vol actualitzar el tauler amb una replanificació de la planificació corresponent al tauler.

Escenari principal:

1. El cap de projecte clica el botó "Replan".
2. La web mostra un missatge preguntant si s'està segur que es vol replanificar.
3. El cap de projecte clica "Acceptar".
4. El plugin actualitza el tauler prèviament seleccionat amb la informació de la replanificació feta per l'eina Replan, crea una card notificació a la llista *Notifications* informant de l'actualització del tauler i les cards finalitzades, com s'han tingut en compte en la replanificació, deixen de tenir accions pendents associades .

Extensions:

3b. El cap de projecte s'adona que no vol replanificar la planificació corresponent al tauler seleccionat.

3b1. El cap de projecte clica "Cancel·lar".

3b2. S'acaba el cas d'ús.

9.1.2. Casos d'ús Trello

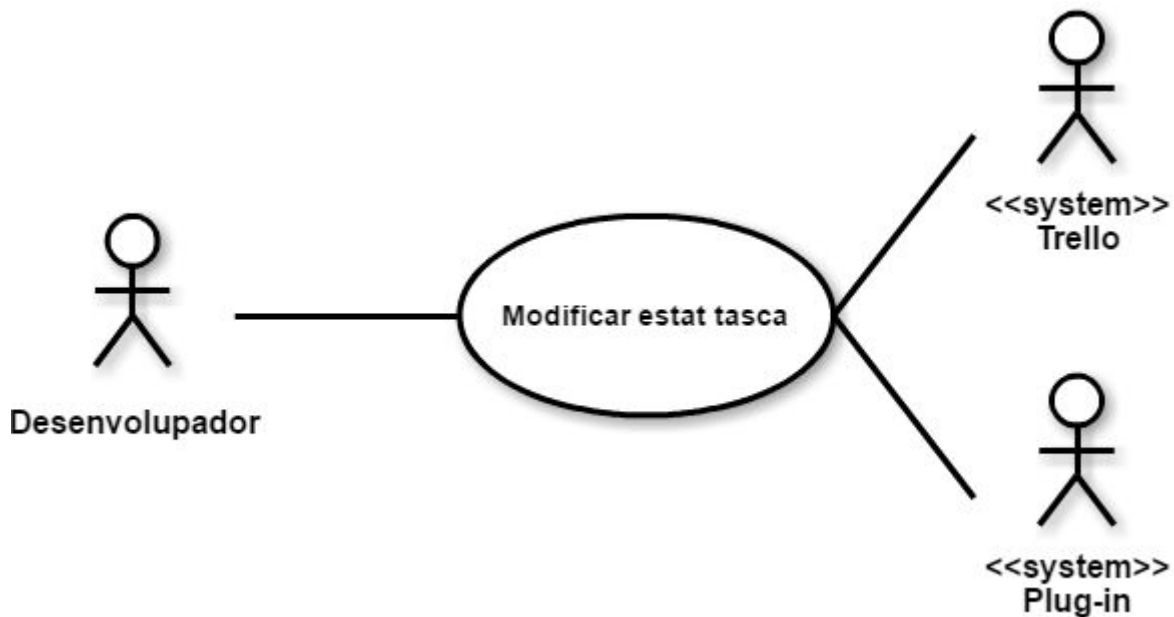


Fig. 6 Diagrama casos d'ús Trello (Font: elaboració pròpia)

9.1.2.1. Modificar estat tasca

Actor principal: Desenvolupador

Precondicions: El desenvolupador ha iniciat sessió a Trello i ha accedit a un tauler on s'ha carregat una planificació mitjançant el plugin.

Disparador: El desenvolupador vol modificar l'estat d'una tasca.

Escenari principal:

1. Es mostra el tauler amb les llistes i cards.
2. El desenvolupador selecciona la card que vol canviar d'estat, l'arrossega i la deixa anar sobre una altra llista vàlida.
3. Es mostra la card canviada de llista i per tant, d'estat
4. Trello informa el plugin d'aquest canvi, que ho enregistra per poder realitzar el seguiment..

Extensions:

2a. El desenvolupador vol moure una card d'una llista a una altra de manera incorrecta.

2a1. El desenvolupador selecciona la card que vol canviar d'estat, l'arrossega i la deixa anar sobre una altra llista no vàlida.

2a2. Es mostra la card canviada de llista.

2a3. Trello informa el plugin d'aquest canvi no permès i torna a moure la card moguda a la llista on estava anteriorment.

2a4. Es mostra la card canviada a la llista on es trobava abans de ser moguda.

9.2. Requisits de qualitat

9.2.1. Requisits de percepció

Requisit #	1	Tipus de requisit	Aparença
Descripció	El disseny de l'aplicació web ha de ser agradable i senzill fet que convidarà a l'usuari a fer-ne ús.		
Justificació	El disseny de l'interfície ha de ser senzill, sense distraccions ni elements cridaners, ja que està aquesta eina està orientada a que sigui utilitzada diàriament per caps de projecte i líders d'equips de treball i no seria convenient que es cansessin del disseny al cap de poc temps.		
Com es durà a terme	S'evitarà utilitzar colors cridaners ni elements carregats de colors.		

Taula 12. Requisit de qualitat aparença (Font: elaboració pròpia)

Requisit #	2	Tipus de requisit	Estil
Descripció	L'aplicació web ha de tenir un estil minimalista i actual.		
Justificació	L'aplicació web ha de ser actual per tal que l'usuari es trobi agust i no es senti allunyat de les tecnologies que utilitza diàriament. Cal fer especial èmfasi en què el disseny ha de ser minimalista evitant sobrecarregar les pàgines amb elements d'interfície de dubtosa utilitat.		
Com es durà a terme	S'utilitzaran estils i elements de Bootstrap els quals s'avenen amb l'estil minimalista i actual buscat. Només s'incorporaran aquells elements que siguin indispensables per utilitzar les funcionalitats de l'aplicació web.		

Taula 13. Requisit de qualitat estil (Font: elaboració pròpia)

9.2.2. Requisits d'usabilitat i humanitat

Requisit #	3	Tipus de requisit	Facilitat d'ús i aprenentatge
Descripció	L'aplicació ha de ser fàcil d'usar per als usuaris.		
Justificació	Malgrat que el sistema està pensat per caps de projectes i líders d'equips que treballen desenvolupant software i aquests usuaris tenen més coneixements que no pas un altre usuari, s'ha d'assegurar que l'aplicació web sigui fàcil d'usar per a qualsevol tipus d'usuari. En tractar-se d'una eina que s'utilitzarà en l'àmbit laboral, com més fàcil d'usar i més fàcil sigui d'aprendre millor.		
Com es durà a terme	S'utilitzaran elements de Bootstrap amb els quals gran part dels usuaris (no només l'usuari que utilitzarà el sistema) estan familiaritzats reduint així el temps d'aprenentatge. Els formularis revisaran que s'hagi introduït o seleccionat la informació necessària abans de ser enviats, minimitzant els errors dels usuaris a l'hora d'introduir les dades. També es donarà a l'usuari feedback de les seves accions tant si s'han completat correctament com sí no.		

Taula 14. Requisit de qualitat facilitat d'ús i aprenentatge (Font: elaboració pròpia)

Requisit #	4	Tipus de requisit	Comprensió i cortesia
Descripció	A l'aplicació web s'utilitzarà un llenguatge clar i concís i respectuós.		
Justificació	L'aplicació no hauria de tenir cap problema de comprensió.		
Com es durà a terme	Els títols, descripcions i etiquetes seran frases clares i concises i s'evitarà incloure grans paràgrafs.		

Taula 15. Requisit de qualitat comprensió i cortesia (Font: elaboració pròpia)

9.2.3. Requisits de rendiment

Requisit #	5	Tipus de requisit	Velocitat i latència
Descripció	El sistema farà totes les seves funcionalitats el més ràpid possible.		
Justificació	És important fer que l'usuari hagi d'esperar el menor temps possible. Si passa el contrari, l'usuari pot quedar desencantat i deixarà d'usar el sistema.		
Com es durà a terme	Es minimitzarà el nombre de crides a servidor tant per l'aplicació web com pel servidor. A l'hora de desenvolupar, es tindrà present l'optimització del codi.		

Taula 16. Requisit de qualitat velocitat i latència

Requisit #	6	Tipus de requisit	Robustesa i tolerància a errors
Descripció	El sistema ha de mostrar robustesa respecte a usos inadequats de les seves funcionalitats per part dels usuaris.		
Justificació	Hauria de complir-se el fet que l'usuari només pugui fer accions que el sistema espera, ja que d'aquesta manera el sistema està preparat per tractar aquestes accions i així evitar qualsevol fallada de sistema.		
Com es durà a terme	Gràcies al control dels formularis i l'ús de desplegable s'assegurarà que la informació introduïda per l'usuari no pugui produir fallades del sistema. També, una vegada l'usuari realitzi una acció que requereixi un temps d'espera, s'impedirà que l'usuari pugui realitzar alguna altra acció que pugui interferir. Finalment, es comprovarà que els desenvolupadors no moguin les targetes d'una llista a una altra de forma incorrecta, desfent l'acció si fes falta.		

Taula 17. Requisit de qualitat robustesa i tolerància a errors (Font: elaboració pròpia)

9.2.4. Requisits operacionals

Requisit #	7	Tipus de requisit	Interacció amb sistemes adjacents
Descripció	El sistema ha de ser capaç d'interaccionar amb Trello i Replan.		
Justificació	En tractar-se d'un sistema que integra l'eina Replan i la plataforma Trello, és de vital importància ser capaç d'interaccionar amb aquests dos sistemes. Sense aquesta interacció el sistema a desenvolupar mancaria de sentit. Documentar-se bé és important per tal d'evitar refer parts del projecte que alentirien el ritme del treball.		
Com es durà a terme	S'estudiarà la documentació disponible dels dos sistemes per tal de saber el seu funcionament, com interaccionar amb ells i quines limitacions tenen. A més a més, es faran crides de prova a les seves APIs tant amb eines externes com amb el sistema a desenvolupar.		

Taula 18. Requisit de qualitat interacció amb sistemes adjacents (Font: elaboració pròpia)

9.2.5. Requisits de manteniment i suport

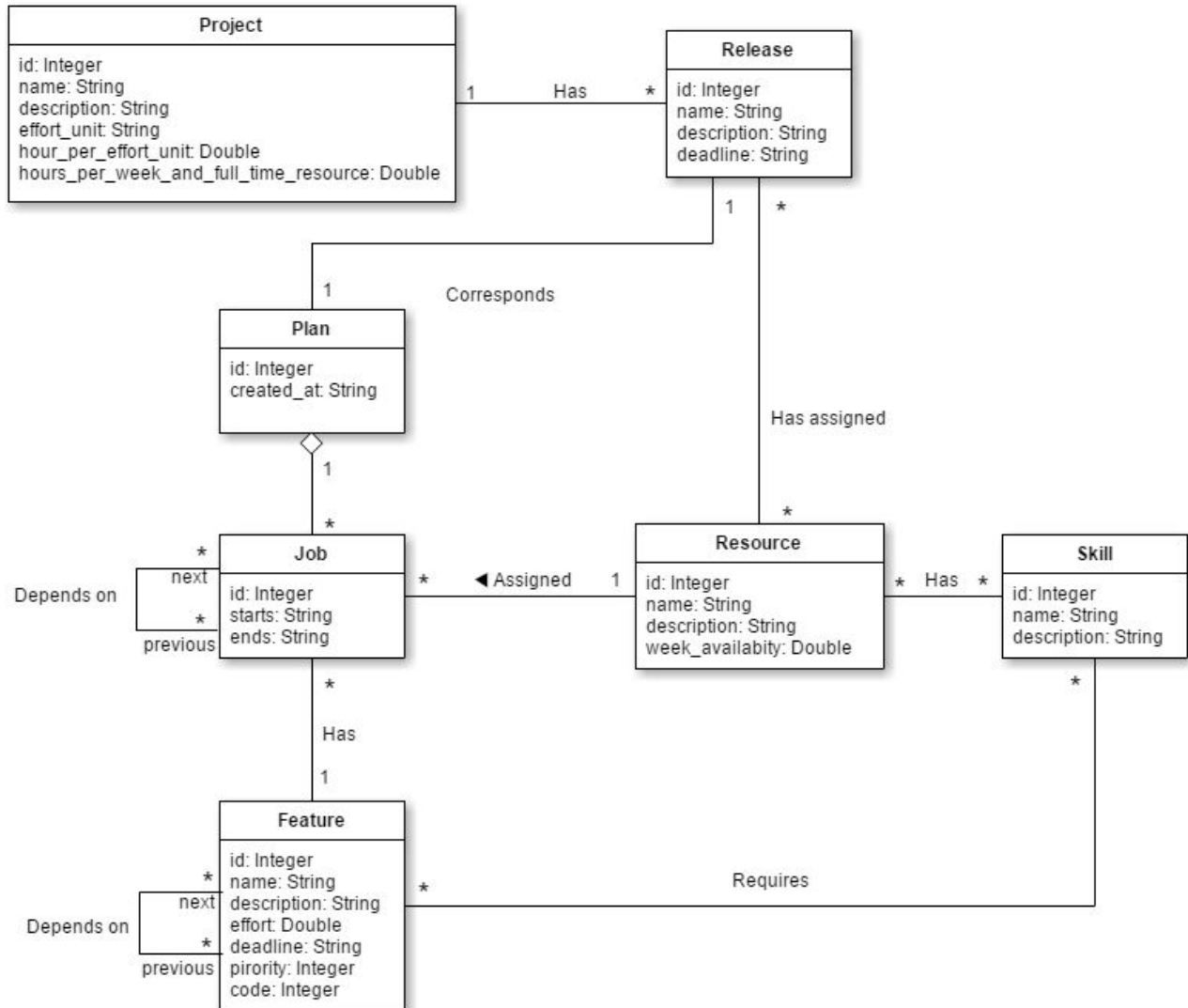
Requisit #	8	Tipus de requisit	Adaptabilitat
Descripció	L'aplicació web serà compatible a través de qualsevol navegador modern.		
Justificació	Es podrà accedir a l'aplicació web a través de les versions actuals dels navegadors més utilitzats: Chrome, Firefox, Edge, Safari, Opera...		
Com es durà a terme	S'utilitzaran llibreries tant d'estils com de scripts que siguin compatibles amb les últimes versions dels navegadors mencionats anteriorment		

Taula 19. Requisit de qualitat adaptabilitat (Font: elaboració pròpia)

Escalabilitat, fiabilitat i disponibilitat, seguretat, legal i capacitat són altres requisits no funcionals que un sistema amb les característiques del sistema a desenvolupar hauria de complir però que en tractar-se d'una prova de concepte i disposar d'un temps limitat, no s'han tingut en compte en el seu desenvolupament.

9.3. Model conceptual

9.3.1. Replan



RI:

- 1) Un Resource no pot estar assignat a un Job que tingui una Feature que requereixi unes Skills que el Resource no té
- 2) Un Resource només pot estar assignat a Jobs que formin part d'un Plan corresponent a una Release a la qual el Resource està assignat
- 3) start < deadline <= ends
- 4) Un Job no pot dependre d'ell mateix
- 5) Una Feature no pot dependre d'ella mateixa

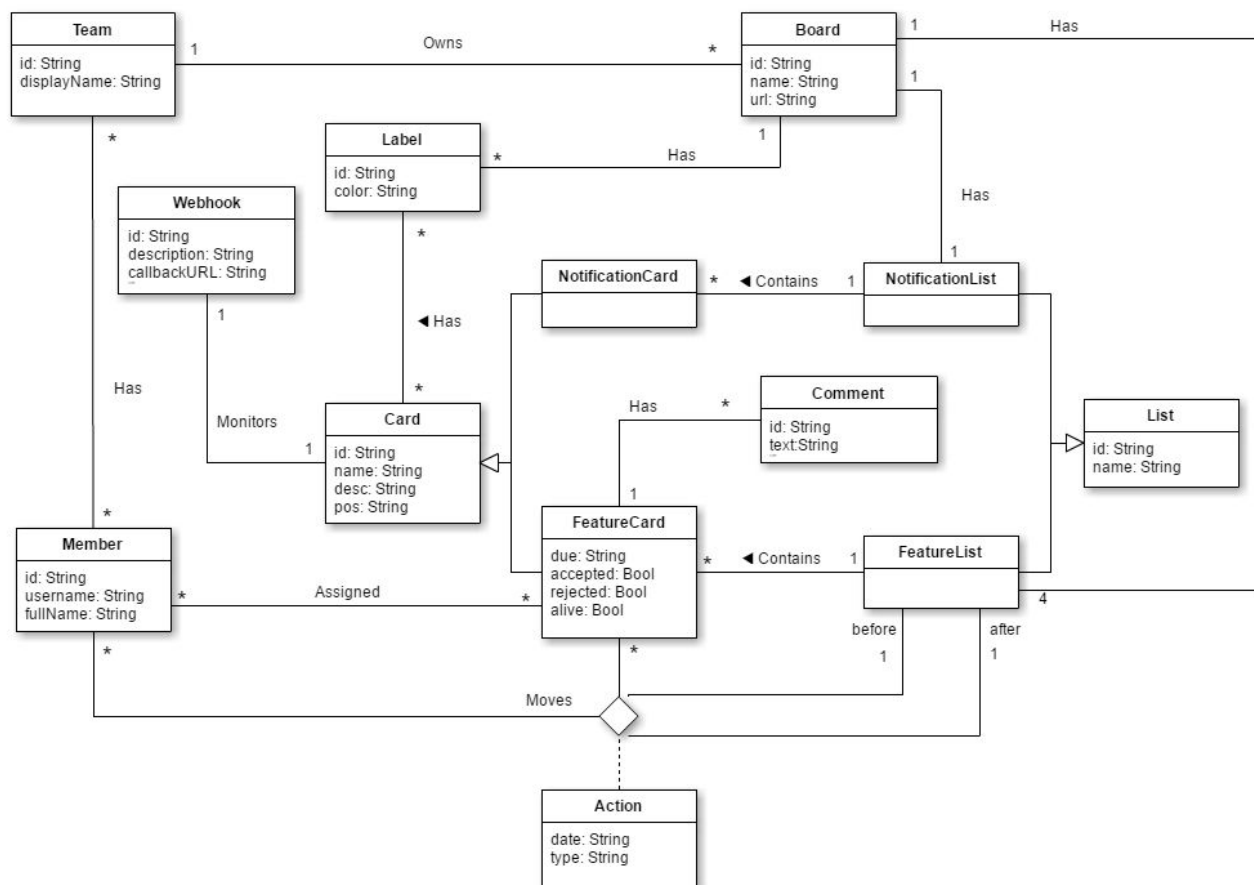
Fig 7. Diagrama conceptual sistema Replan (Font: elaboració pròpia)

A partir de la documentació de l'eina Replan i de la informació que retornen les crides a l'API del Replan Controller, s'ha elaborat aquest diagrama que mostra els conceptes relacionats amb les planificacions que genera l'eina Replan.

La planificació generada es troba representada pel concepte plan. Un plan és una planificació d'una release generada per Replan. Aquest plan està format per jobs. Al seu torn, cada job relaciona una tasca (feature) amb el recurs que l'ha de realitzar (resource). Els jobs poden dependre de la finalització d'altres jobs perquè aquest puguin ser realitzats. El mateix passa amb les features, que poden dependre de la finalització d'altres features per a aquestes puguin ser realitzades.

Malgrat que el concepte de skill no es té en compte a l'hora de desenvolupar el projecte, és important que aparegui en aquest diagrama, ja que ajuda a entendre millor com funcionen l'eina Replan a l'hora de generar les planificacions. Per realitzar una feature es requereixen unes skills específiques, per tant els recursos que siguin assignats a aquesta feature han de tenir aquestes skills.

9.3.2. Trello



RL:

- 1) Les FeatureCards de la llista FeatureList amb name = On-hold només poden ser mogudes a la FeatureList amb name = Ready
- 2) Les FeatureCards de la llista FeatureList amb name = Ready només poden ser mogudes a la FeatureList amb name = In Progress
- 3) Les FeatureCards de la llista FeatureList amb name = In Progress només poden ser mogudes a la FeatureList amb name = Done
- 4) Les FeatureCards de la llista FeatureList amb name = Done no poden ser mogudes a una altra FeatureList

Fig 8. Diagrama model conceptual sistema Trello

En aquest diagrama es poden veure els conceptes de Trello relacionats amb el projecte. La majoria dels conceptes representats ja s'han vist a l'apartat *Representació de la planificació a Trello*.

Primer de tot, cal diferenciar entre les cards que corresponen a una feature (**featureCard**) i les cards el propòsit de les quals és informar els usuaris que treballen amb aquest board (**notificationCard**). També és important mencionar que aquesta mateixa diferenciació es fa també en les **lists** que contenen les cards per tal de fer més clar que les notificationCards no es poden moure a una altra list.

Per tal de monitoritzar els canvis en l'estat de les cards és imprescindible associar un **webhook** a cada **card**, per aquest motiu s'ha estimat representar aquest concepte en aquest diagrama.

Les **labels** són etiquetes de colors que es poden posar a les cards. Cada board té unes labels predifinides que ofereix el mateix Trello. Tot i ser predifinides i sempre dels mateixos colors, les labels són úniques per a cada **board** a causa de la seva id que les identifica.

Les restriccions d'integritat s'han afegit per evitar que els usuaris facin un ús incorrecte de les lists de Trello.

Els noms utilitzats són els propis que utilitza Trello en el seu sistema a excepció dels conceptes featureCard, featureList, notificationCard i notificationList que no són originals de Trello, sinó que són conceptes propis d'aquest projecte.

9.3.3. Plugin i relacions entre sistemes

Els conceptes propis del plugin a desenvolupar no tenen molt de sentit per si sols, ja que la seva importància depèn de les seves relacions amb els conceptes dels sistemes a integrar, és a dir, amb els conceptes de Trello i Replan.

En els diagrames següents es poden veure els conceptes dels tres sistemes implicats. Els conceptes del Replan són de color groc, els del Plugin de color verd i els de Trello de color blau.

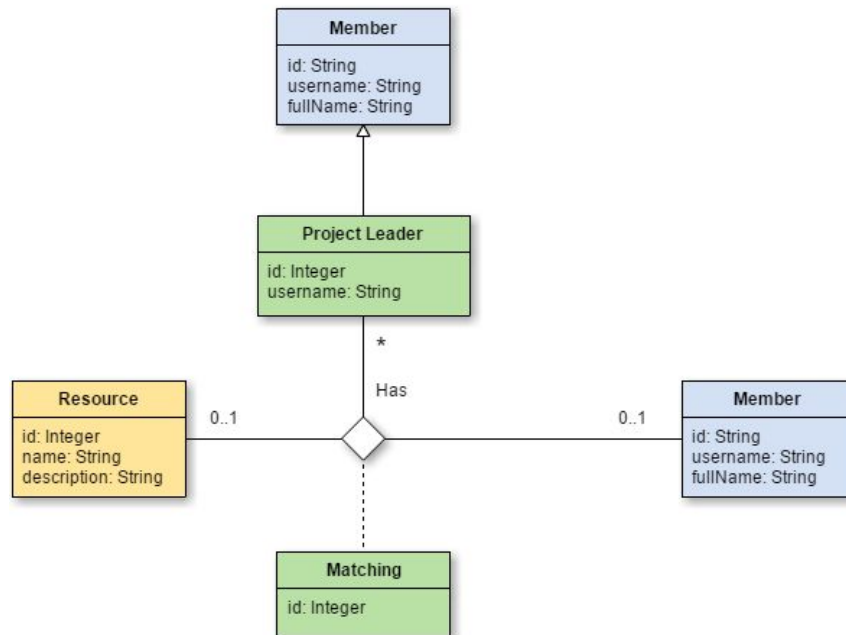


Fig 9. Diagrama model conceptual Plugin associació de recursos (Font: elaboració pròpia)

Aquest diagrama mostra el concepte de matching, concepte resultant de la relació entre el cap de projecte (**Project Leader**), un recurs de la planificació (**resource**) i un usuari de Trello (**member**). El concepte matching sorgeix de la necessitat de relacionar un recurs amb el seu usuari a Trello per tal de poder treballar amb la planificació a Trello correctament. Els matchings no estan lligats a un sol tauler, ja que aquest concepte està pensat perquè el cap de projecte pugui reutilitzar aquestes associacions entre recursos i membres quan vulgui carregar una altra planificació a Trello, estalviant-se haver de tornar a associar recursos i membres quan ja ho havia fet anteriorment.

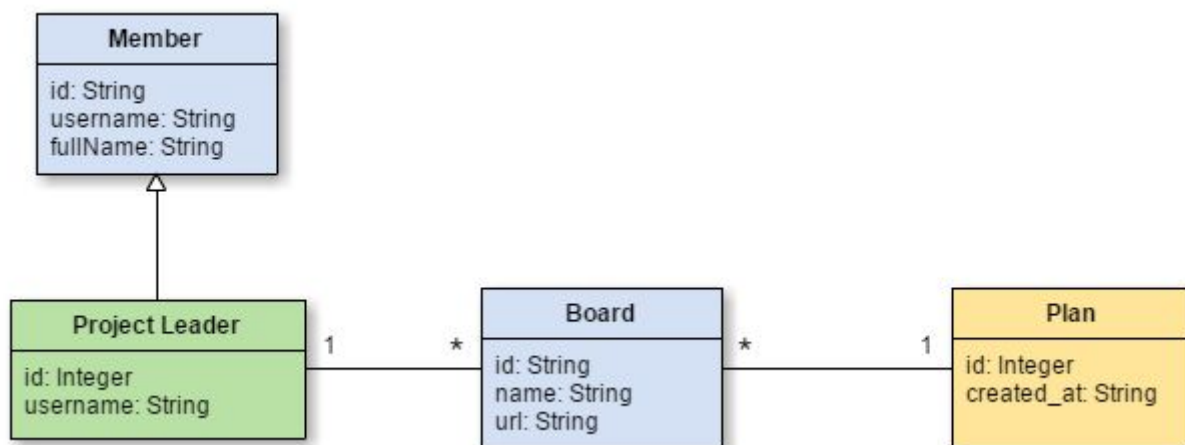


Fig 10. Diagrama model conceptual relació Board-Plan (Font: elaboració pròpia)

Aquest diagrama mostra com el cap de projecte (**Project Leader**) pot tenir diferents boards, boards els quals representen una planificació generada per l'eina Replan (**plan**).

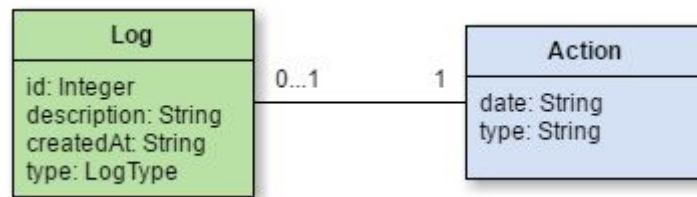


Fig 11. Diagrama model conceptual relació Log-Action (Font: elaboració pròpia)

En aquest esquema es mostra com una **action** de Trello correspon a un **log** propi del plugin. Els logs permeten diferenciar els moviments de cards entre llistes, que són un tipus concret d'acions que tenen lloc a Trello, gràcies a l'atribut de tipus type, que és de tipus LogType, que és una enumeració que es detallarà més endavant.

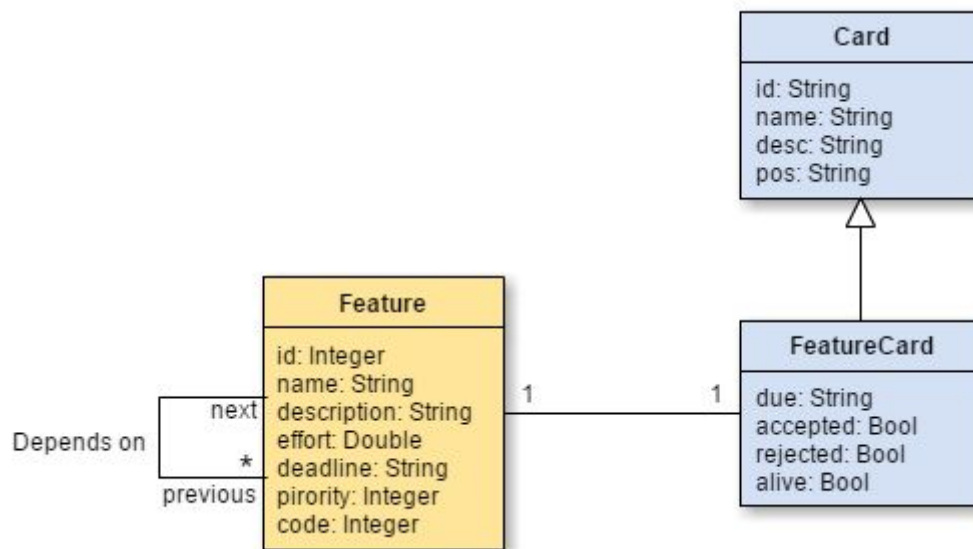


Fig 12. Diagrama model conceptual FeatureCard-Card

Com ja s'ha comentat en l'apartat *Representació de la planificació a Trello*, el concepte de **feature** de l'àmbit del Replan es representa com a **card** a Trello, més concretament en una **featureCard**.

10. Desenvolupament del sistema

10.1 Front-end del Plugin

El front-end és una SPA (Single Page Application) implementada amb els *frameworks* AngularJS i Bootstrap 3.

AngularJS és un *framework* MVC (model, vista, controlador) en JavaScript que permet crear aplicacions SPA^[15]. Els components bàsics d'una aplicació AngularJS són els següents:

- **Fitxer `index.html`:** aquest fitxer conté el html comú per totes les vistes i és on es mostraran les diferents vistes de l'aplicació.
- **Vistes:** fitxers html que contenen el html propi de cada pantalla de l'aplicació.
- **Controladors:** fitxers html que contenen la lògica de l'aplicació de la part de front-end.
- **Mòdul:** fitxer JavaScript comuntment anomenat `app.js` que funciona de contenidor pels diferents components de l'aplicació (controladors, serveis ...). També gestiona l'enrutament tot vinculant una ruta de la web amb un controlador i una vista.

Adicionalment, per aquesta aplicació s'ha fet ús de serveis, que són les classes encarregades de realitzar la comunicació amb el back-end de l'aplicació.

En AngularJS no es manipula el DOM de la pàgina en els controladors, ja que aquesta manipulació té lloc a les vistes a través de l'ús de directives. AngularJS compta amb una gran quantitat de directives que ja venen donades (com per exemple `ng-repeat` o `ng-model`) que faciliten en gran mesura la manipulació del DOM.

Les vistes es comuniquen amb els controladors associats a través del que es coneix com a **scope** (`$scope`) a partir del qual es té accés al context de l'aplicació. Per la seva banda els controladors es comuniquen amb els serveis gràcies a les injeccions de dependència

^[16].

També es pot compartir el context general de l'aplicació independentment de la vista actual a través del **root scope** (`$rootScope`). En els diagrames de disseny intern que es veuran en aquesta secció, **Session** és l'equivalent al root scope.

La figura següent il·lustra els components i la interacció entre ells:

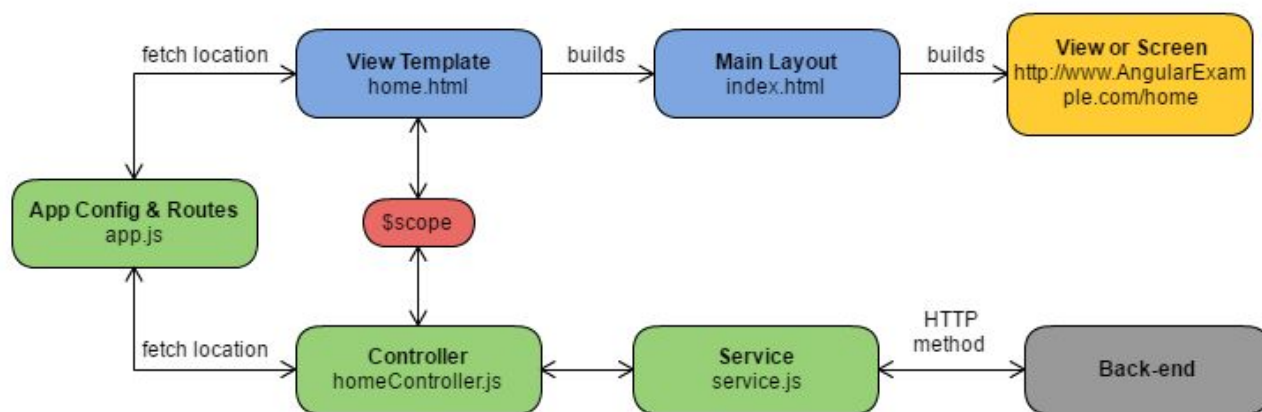


Fig 13. Components bàsics d'una aplicació d'AngularJS (Font: elaboració pròpia)

10.1.1. Diagrama de navegabilitat

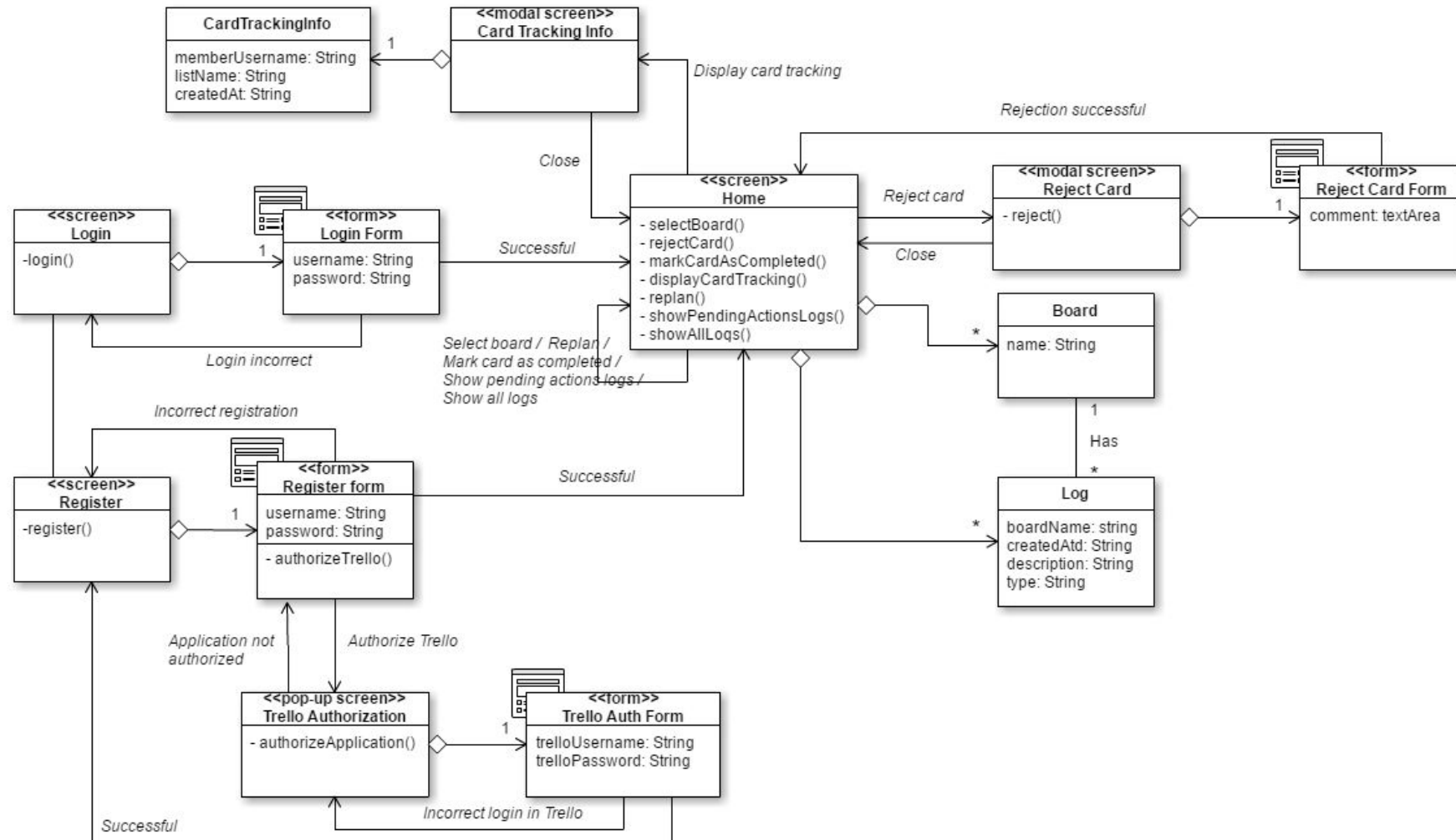
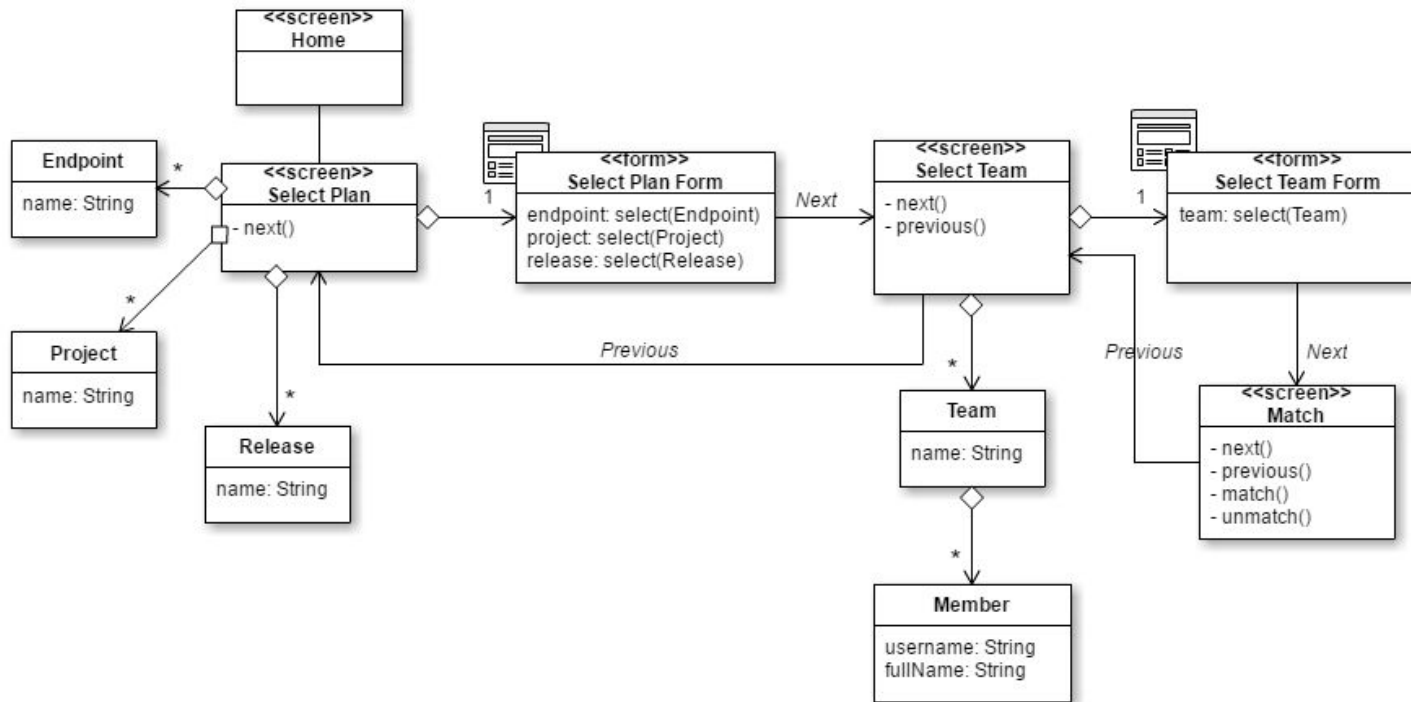


Fig 14. Diagrama de navegabilitat (login, registre i pàgina home) (Font: elaboració pròpia)



Notes:

- 1) S'ha dividit el diagrama en dues parts per qüestions d'espai. Els diagrames es connecten a través de la pàgina home.
- 2) Des de qualsevol de les pàgines es pot anar a la pàgina home a través dels enllaços del menú de navegació present en tota la pàgina (excepte en les pàgines de login i registre).

Fig 15. Diagrama de navegabilitat (procés de carregar planificació a Trello 1a part) (Font: elaboració pròpia)

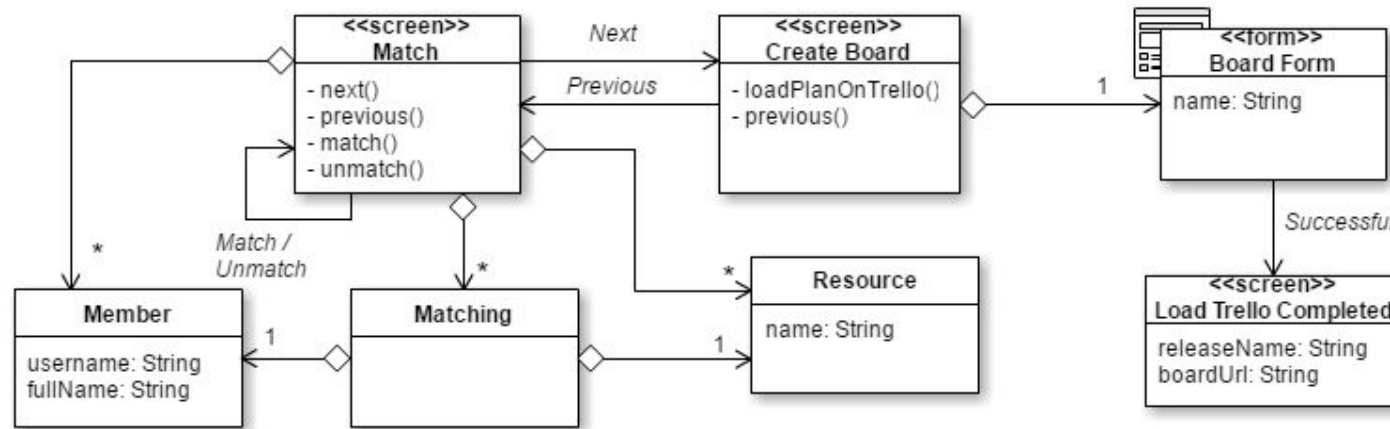


Fig 16. Diagrama de navegabilitat (procés de carregar planificació a Trello 2a part) (Font: elaboració pròpia)

10.1.2. Diagrama disseny intern

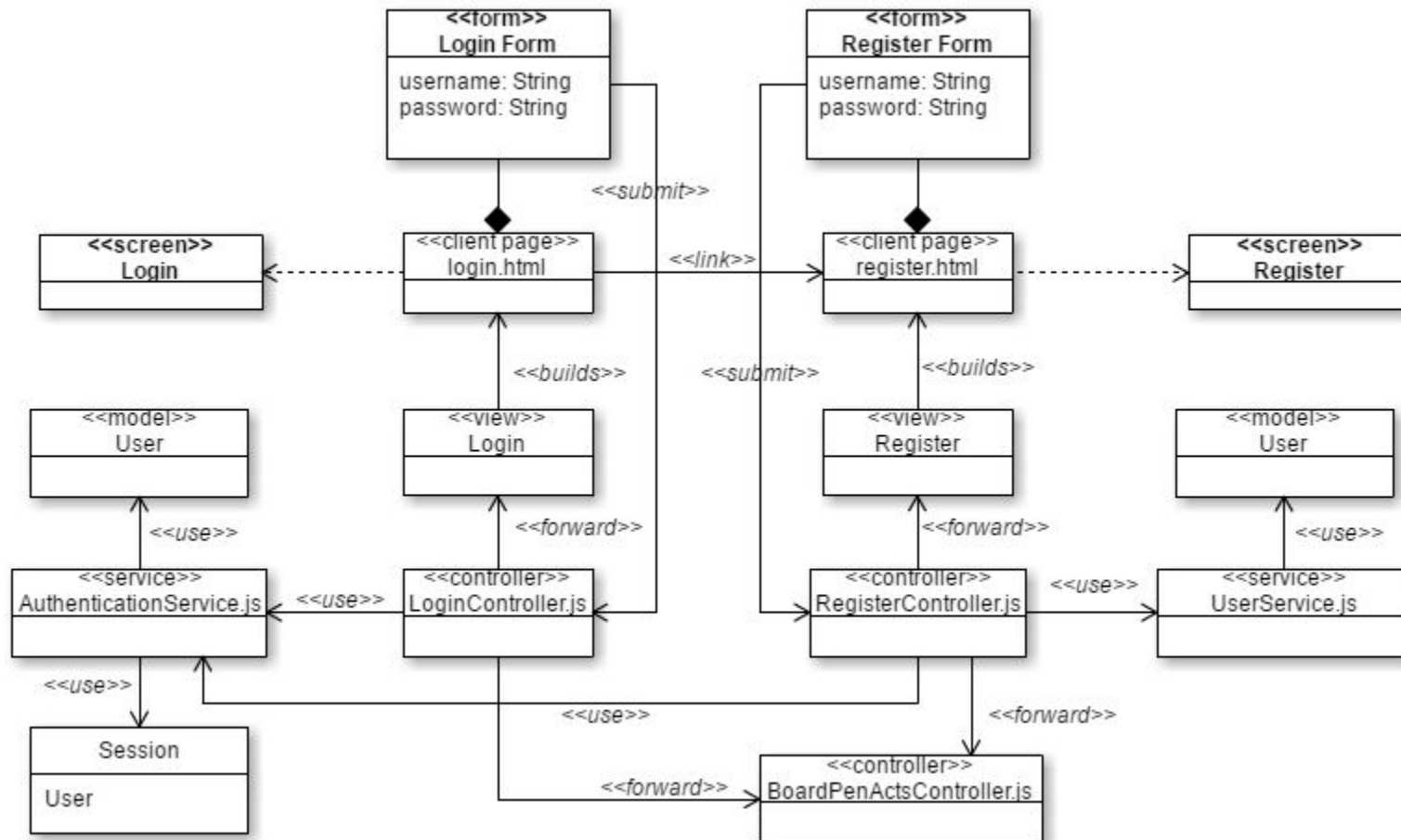


Fig 17. Diagrama disseny intern (login i registre) (Font: elaboració pròpia)

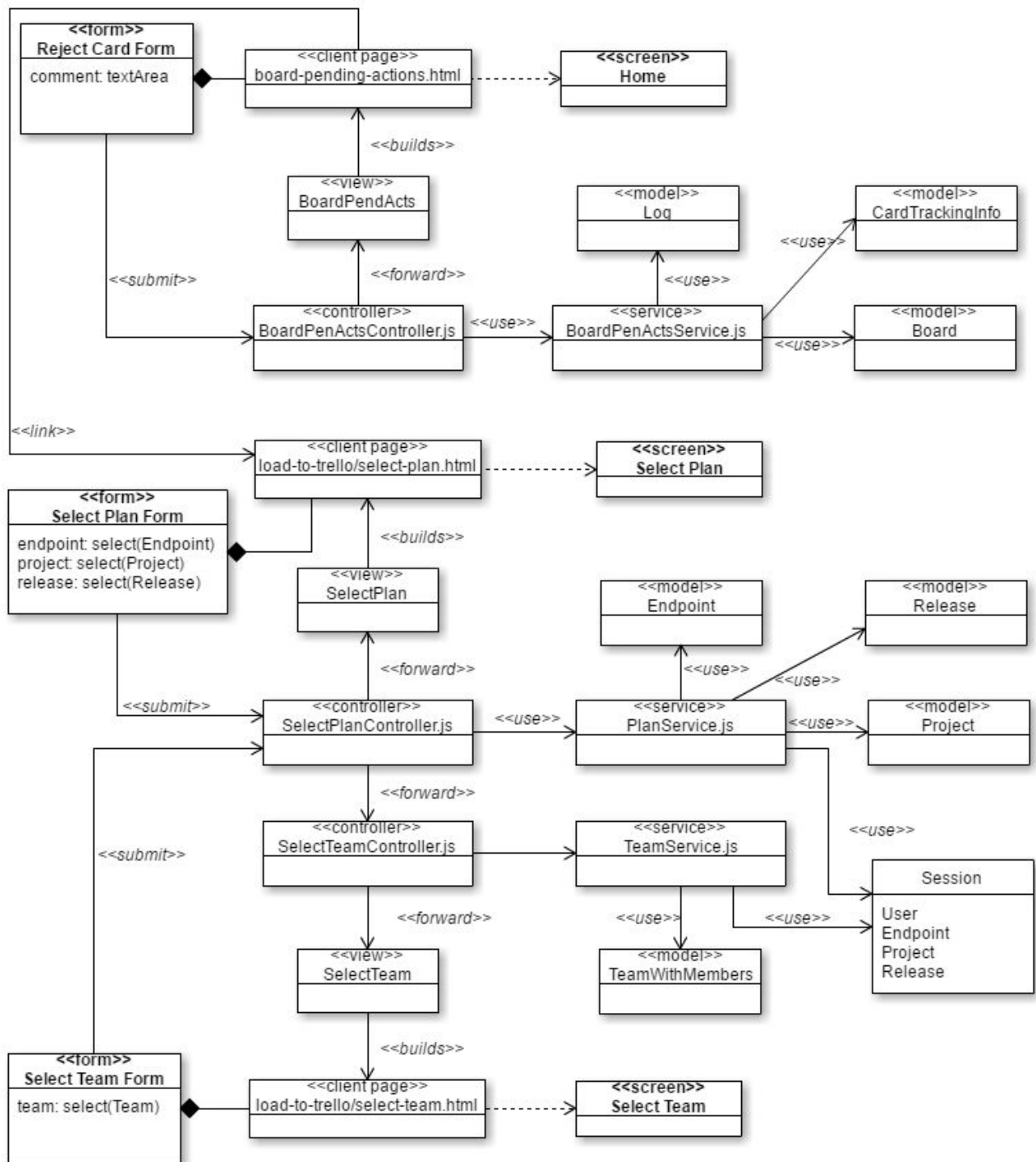


Fig 18. Diagrama disseny intern (refusar card i primeres 2 pàgines del procés de carregar planificació a Trello) (Font: elaboració pròpia)

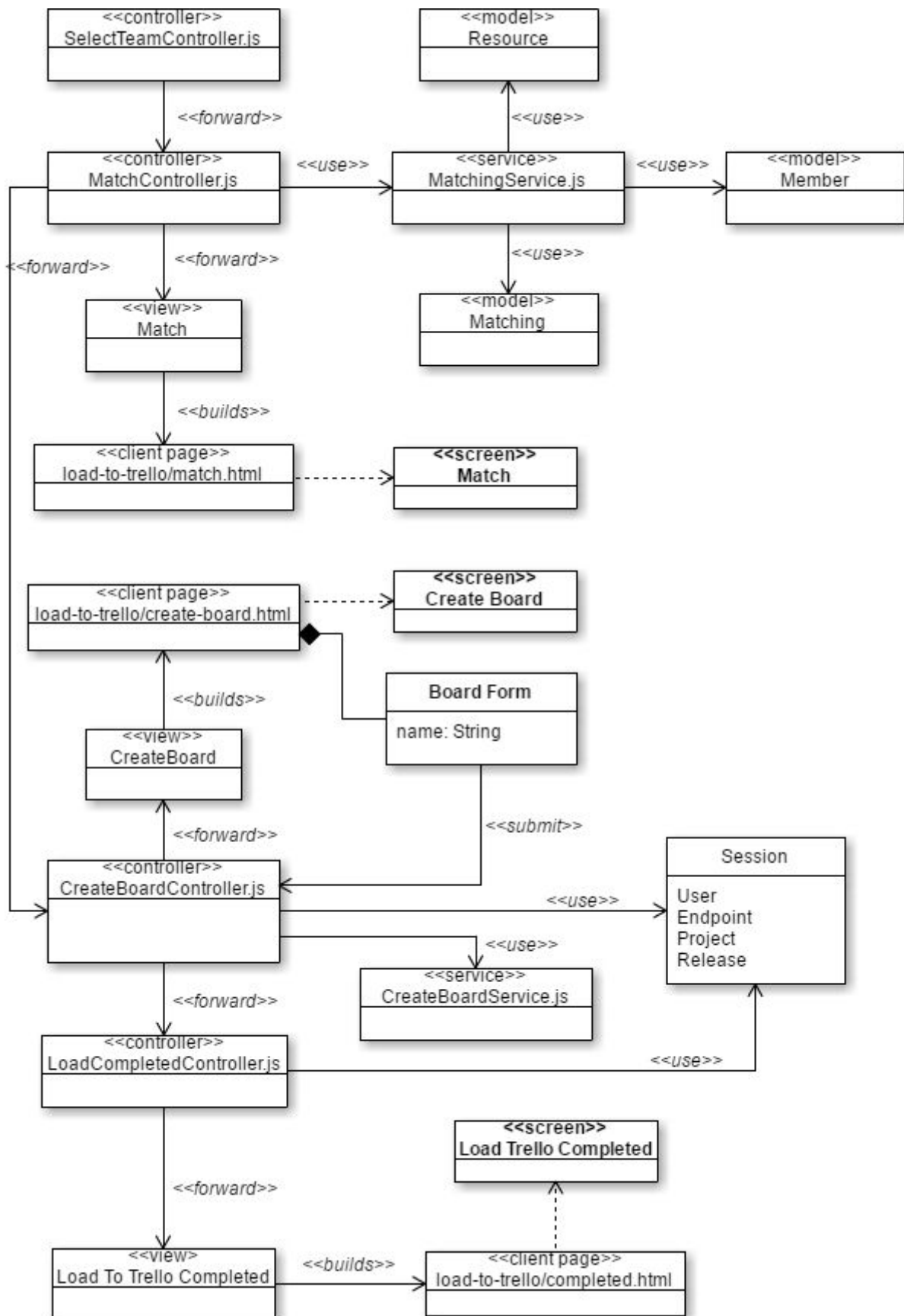


Fig 19. Diagrama disseny intern (darreres 3 pàgines del procés de carregar planificació a Trello) (Font: elaboració pròpia)

En tractar-se d'una single page application feta amb AngularJS totes les client pages que es poden veure en aquests diagrames estan contingudes dins de l'**index.html**.

10.2 Back-end del Plugin

El back-end del plugin s'ha implementat amb Spring Boot. Per la persistència s'utilitza Hibernate i Spring JPA Data.

Spring Boot és un *framework* de Java que facilita la creació d'aplicacions Spring. Les aplicacions creades amb aquest *framework* tenen Tomcat incrustat, fet que permet que puguin ser ràpidament desplegades sense gaire configuració.

Per tal de poder implementar les funcionalitats especificades, el back-end del plugin ha de ser capaç de:

- Consumir serveis externs (APIs de Trello i Replan controller).
- Exposar una API perquè sigui consumida tant pels sistemes externs com pel front-end del plugin.
- Emmagatzemar dades de l'usuari de l'aplicació web, la informació de les planificacions utilitzades per crear taulers a Trello i la informació d'aquests taulers.

Per consumir serveis externs s'utilitza **RestTemplate**, una classe del Spring base. Per exposar una API, s'utilitza **Spring Boot Actuator**, un subprojecte de Spring Boot que permet que la creació de RESTful Web Services. Tant en les peticions entrants com en les peticions realitzades a serveis externs, s'utilitza **Jackson**, una llibreria que incorpora Spring que fa la conversió de JSON a classes Java. Finalment, per la persistència s'utilitzen les classes de **Spring Data JPA**: les interfícies **Data Repositories** per emmagatzemar i accedir a dades, **JPA Entities** per la representació de models de domini i **Hibernate** per fer el mapeig objecte-relacional (ORM).

La següent figura pretén donar una visió molt general dels components bàsics que tindria una aplicació amb característiques similars a les del back-end del plugin.

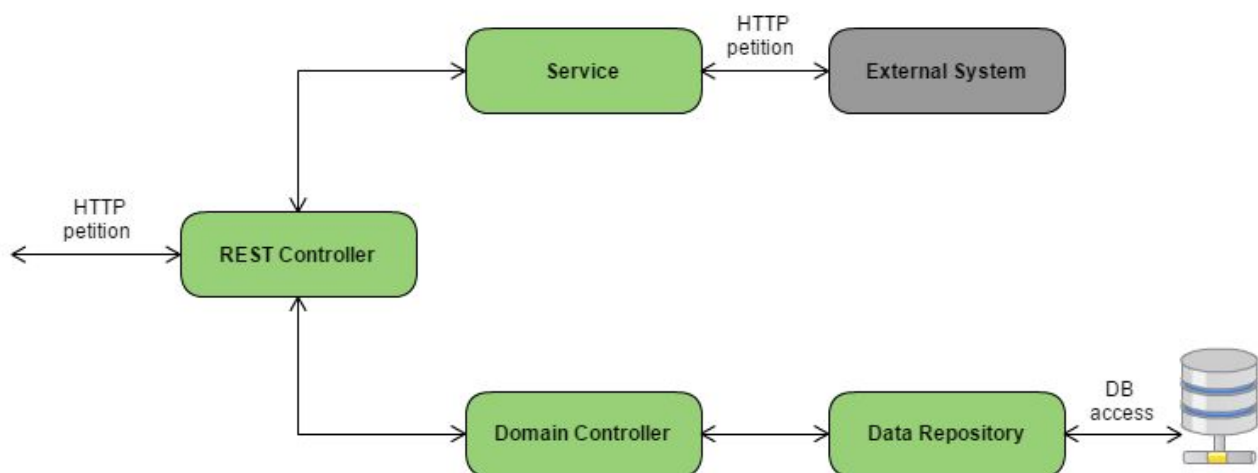


Fig 20. Esquema general components aplicació similar al back-end del plugin (Font: elaboració pròpia)

En aquesta imatge es pot veure com un REST Controller s'encarrega de gestionar les peticions que rep el servidor. Depenent de la petició pot necessitar tractar amb dades bé siguin locals o que té un servei extern i fins i tot dels dos tipus. El primer dels casos, es veuria representat pel camí inferior i el segon pel camí superior.

A part d'aquests components, per tal d'iniciar el servidor cal una classe amb l'anotació `@SpringBootApplication` que executi el mètode `run`. També, per poder inicialitzar la persistència, s'ha creat una classe amb l'anotació `@Component` que inicialitza els repositoris.

10.2.1. Diagrama classes back-end

A l'esquema següent es pot veure les diferents capes que formen la part de servidor i com interactuen entre elles. Les classes que hi apareixen són classes d'exemple per tal d'aclarir la interacció entre elements de diferents capes.

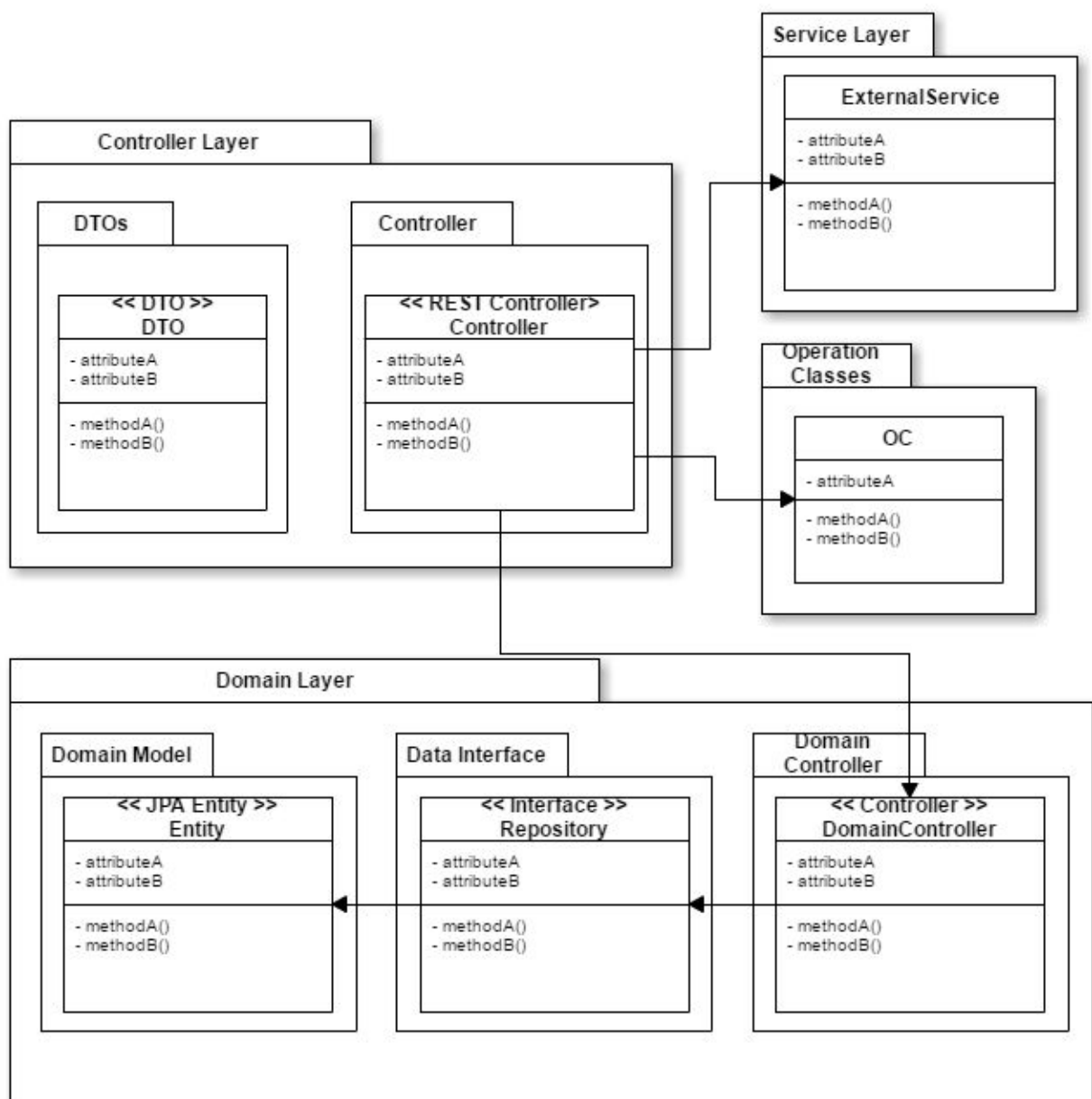


Fig 21. Diagrama de les capes del backend (Font: elaboració pròpia)

10.2.1.1 Diagrames classes capa controladors

La cap de controladors conté les classes DTO i els controladors d'operacions REST. Les classes DTO són utilitzades en l'intercanvi de dades entre el servidor i els sistemes externs a integrar (Replan i Trello) a través de les seves APIs. També són utilitzades en la comunicació amb la part de client, és a dir, la web. D'altra banda, els controladors s'encarreguen de rebre les peticions http i donar resposta a aquestes.

Classes DTO

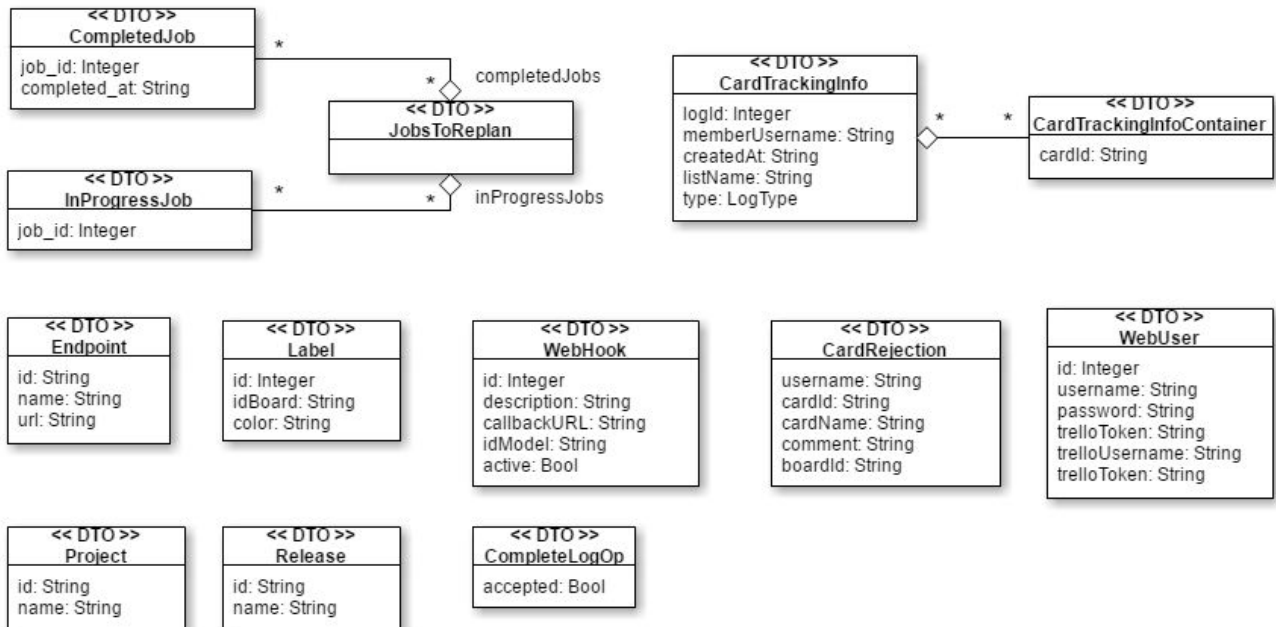


Fig 22. Diagrama classes DTO 1 (Font: elaboració pròpia)

Nota: els constructors, getters i setters s'han obviat

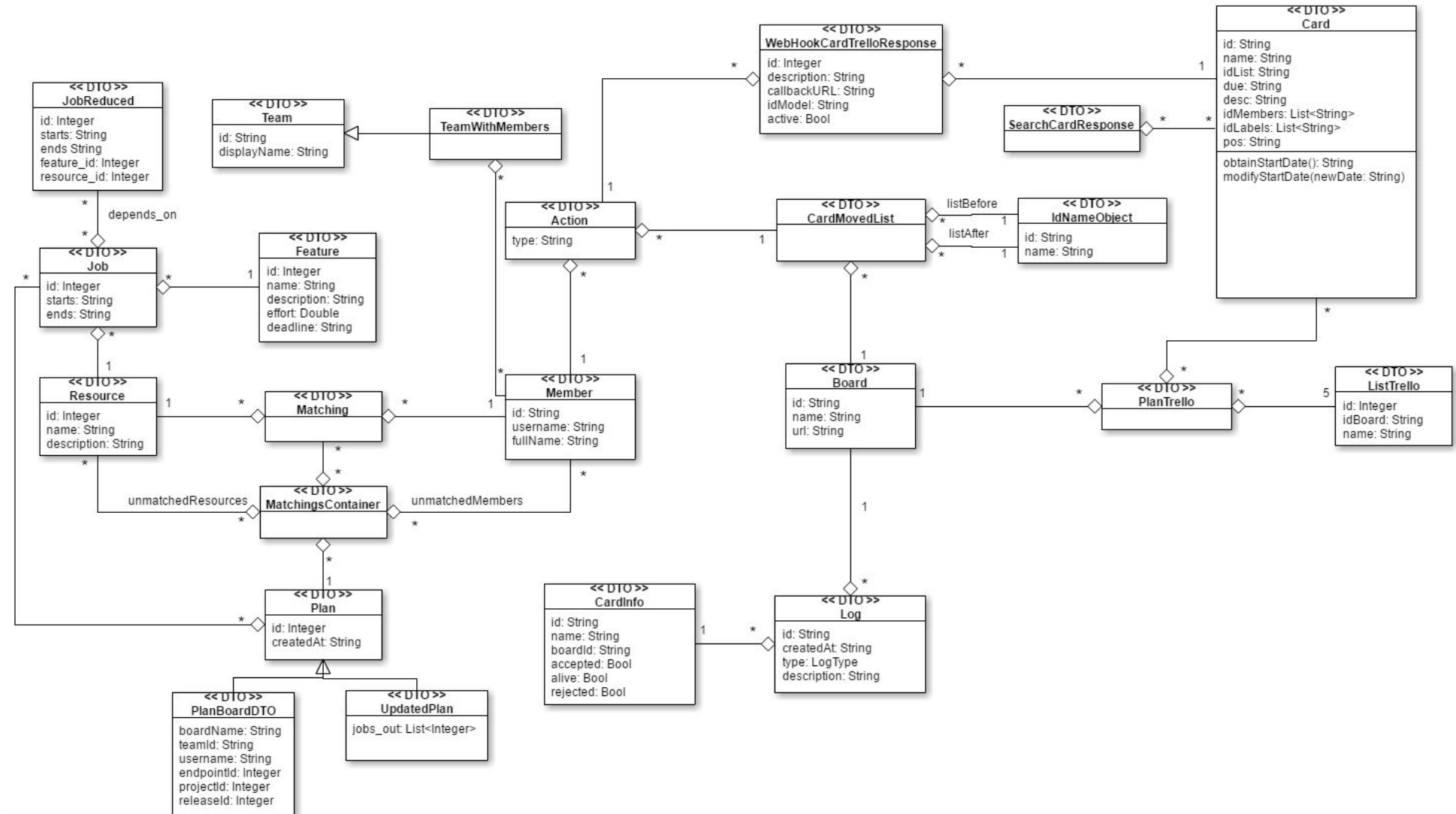


Fig 23. Diagrama classes DTO 2 (Font: elaboració pròpia)

Nota: els constructors, getters i setters s'han obviat

Controllers

<< REST Controller >> AuthenticationController
createUser(authInfo: Map<String,String>): ResponseEntity<User>

<< REST Controller >> BoardsController
createBoard(planBoardDTO: PlanBoardDTO): ResponseEntity<PlanTrello> getBoards(username: String): ResponseEntity<set(Board)>

<< REST Controller >> CardsController
getCardTrackingInfo(cardId: String): ResponseEntity<CardTrackingInfoContainer> rejectCard(cardRejection: CardRejection): ResponseEntity<CardInfo> changeAcceptedCard(String: cardId): ResponseEntity<CardInfo>

<< REST Controller >> LogsController
getLogs(username: String, boardId: String): ResponseEntity<set(Log)> doReplan(boardId: String): ResponseEntity<String> doReplanFake(boardId: String): ResponseEntity<set(Card)> getNextCard(cardsUpdated: set(Card), boardId: String, readyListId: String): set(Card)

<< REST Controller >> MatchingsController
createMatchings(username: String, newMatchings: set(Matching): ResponseEntity<set(Matching)> deleteMatchings(username: String, matchingsToDelete: set(Matching): ResponseEntity<set(Matching)> getMatchings(username: String, endpointId: Integer, projectId: Integer, releaseId: Integer, teamId: String): ResponseEntity<MatchingsContainer>

<< REST Controller >> ProjectsController
getEndpoints():ResponseEntity<set(Endpoint)> getProjects(endpointId: String): ResponseEntity<set(Project)> getReleases(endpointId: String, projectId: String): ResponseEntity<set(Release)>

<< REST Controller >> TeamsController
getTeams(username: String):ResponseEntity<set(Teams)> getTeamMembers(username: String, teamId: String): ResponseEntity<set(Member)>

<< REST Controller >> TrelloCallbacksController
checkWebhook(username: String):ResponseEntity<HttpStatus> cardModified(username: String): ResponseEntity<set(String)> getNextCard(cardsAssigned(set(Card), readyListId: String, inProgressListId: String, onHoldListId: String): Card getReleases(endpointId: String, projectId: String): ResponseEntity<set(Release)> createLog(cardId: String, cardName: String, memberUsername: String, logType: LogType): cardHasLabel(idLabel: String, idLabels: set(String): Bool

<< REST Controller >> UsersController
createUser(user: WebUser): ResponseEntity<WebUser>

Fig 24. Diagrama classes controladors (Font: elaboració pròpia)

10.2.2.2 Diagrama classes capa de serveis



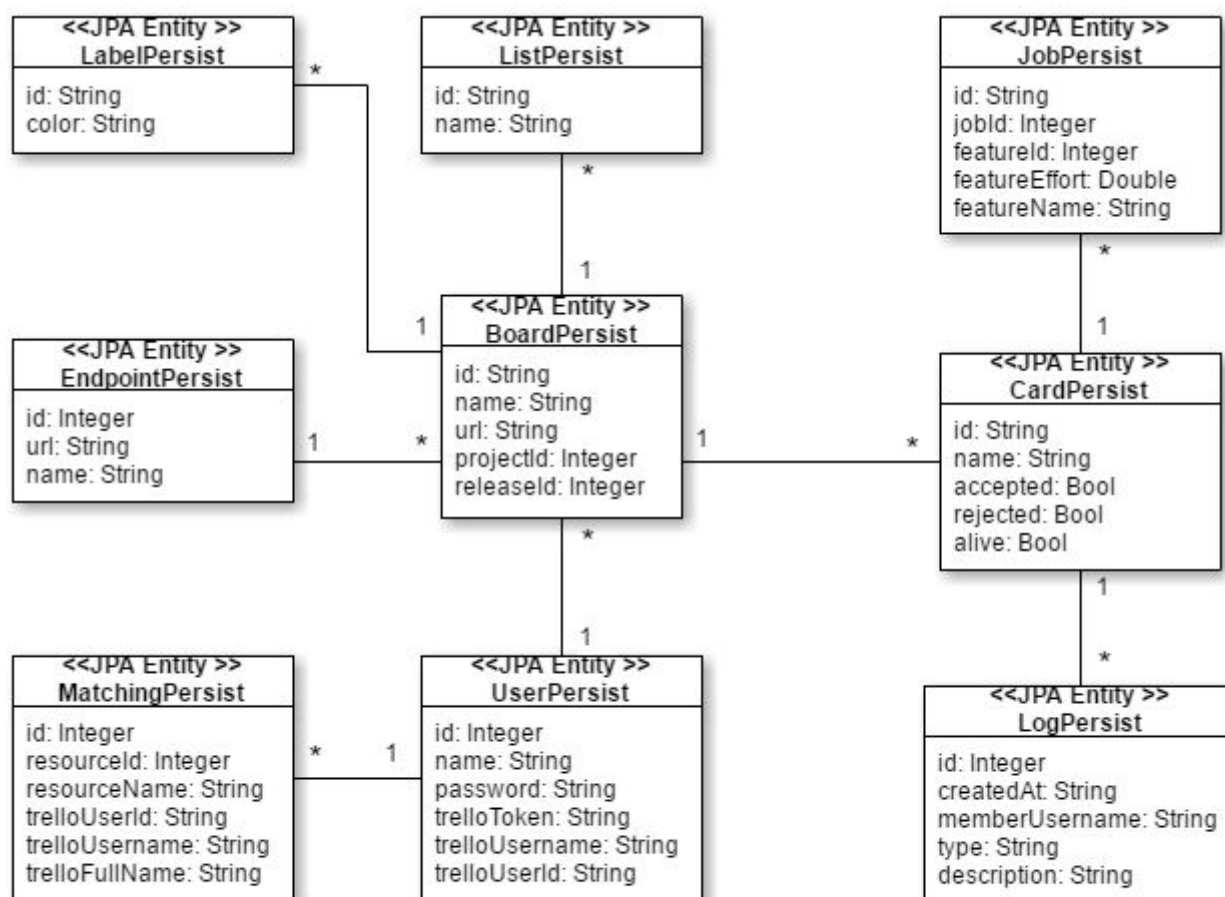
Fig 25. Diagrama classes serveis (Font: elaboració pròpia)

Les classes d'aquesta capa duen a terme la comunicació de la part de servidor amb els sistemes externs mitjançant peticions a les APIs que aquests exposen.

La classe **TrelloService** fa ús d'algunes de les operacions que exposa l'API de Trello. Per la seva banda, la classe **ReplanService** interactua amb l'eina Replan a través de l'API que exposa el Replan Controller, un dels components que formen l'eina Replan. En canvi, **ReplanFakeService** és una classe creada exclusivament per la replanificació. El seu propòsit és simular la interacció amb l'API de Replan retornant unes planificacions predefinides per tal de poder ser replanificades. Aquestes replanificacions també són predefinides per tal de poder mostrar diferents situacions preparades.

10.2.2.3 Diagrames classes capa de domini

Domain Model



RI:

Per un mateix usuari no pot haver-hi més d'un matching amb la mateixa combinació de trelloUserId i resourceId

Fig 26. Diagrama classes JPA (Font: elaboració pròpia)

Aquest diagrama mostra les classes JPA a partir de les quals Hibernate generarà les taules. Les classes JPA corresponen als conceptes i informació que és imprescindible guardar a persistència o que tenir-ho a persistència suposa un estalvi de crides a serveis externs.

Per tal d'optimitzar les consultes a base de dades i reduir el nombre joins, **MatchingPersist** conté la informació necessària tant del recurs com del membre.

Les labels i les llistes es guarden per reduir el nombre crides de serveis externs, ja que la seva informació no canvia.

Domain Model (classes per simular Replan)

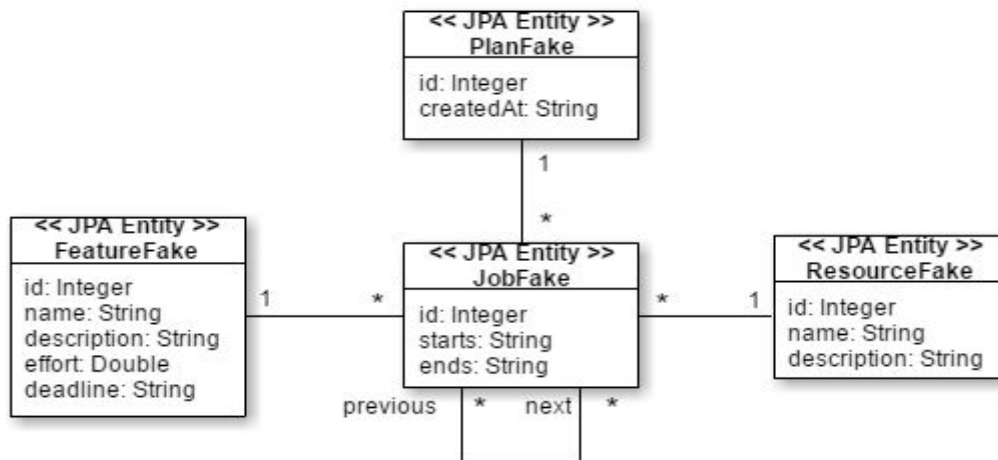
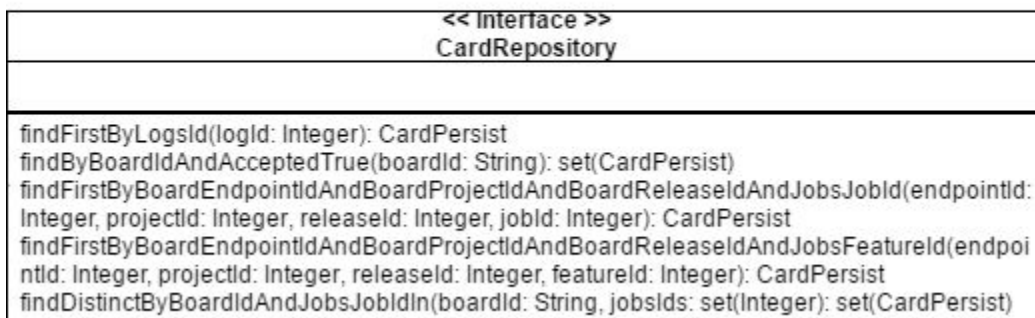
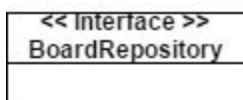


Fig 27. Diagrama classes JPA per simular Replan (Font: elaboració pròpia)

Aquestes classes JPA s'han creat amb l'únic propòsit de guardar aquesta informació per fer replanificacions. Aquesta informació la té el Replan i per tant, si no se simulés el Replan no caldria tenir-la a persistència, ja que la informació necessària respecte als plans es mostra al diagrama anterior a aquest.

Data Interface



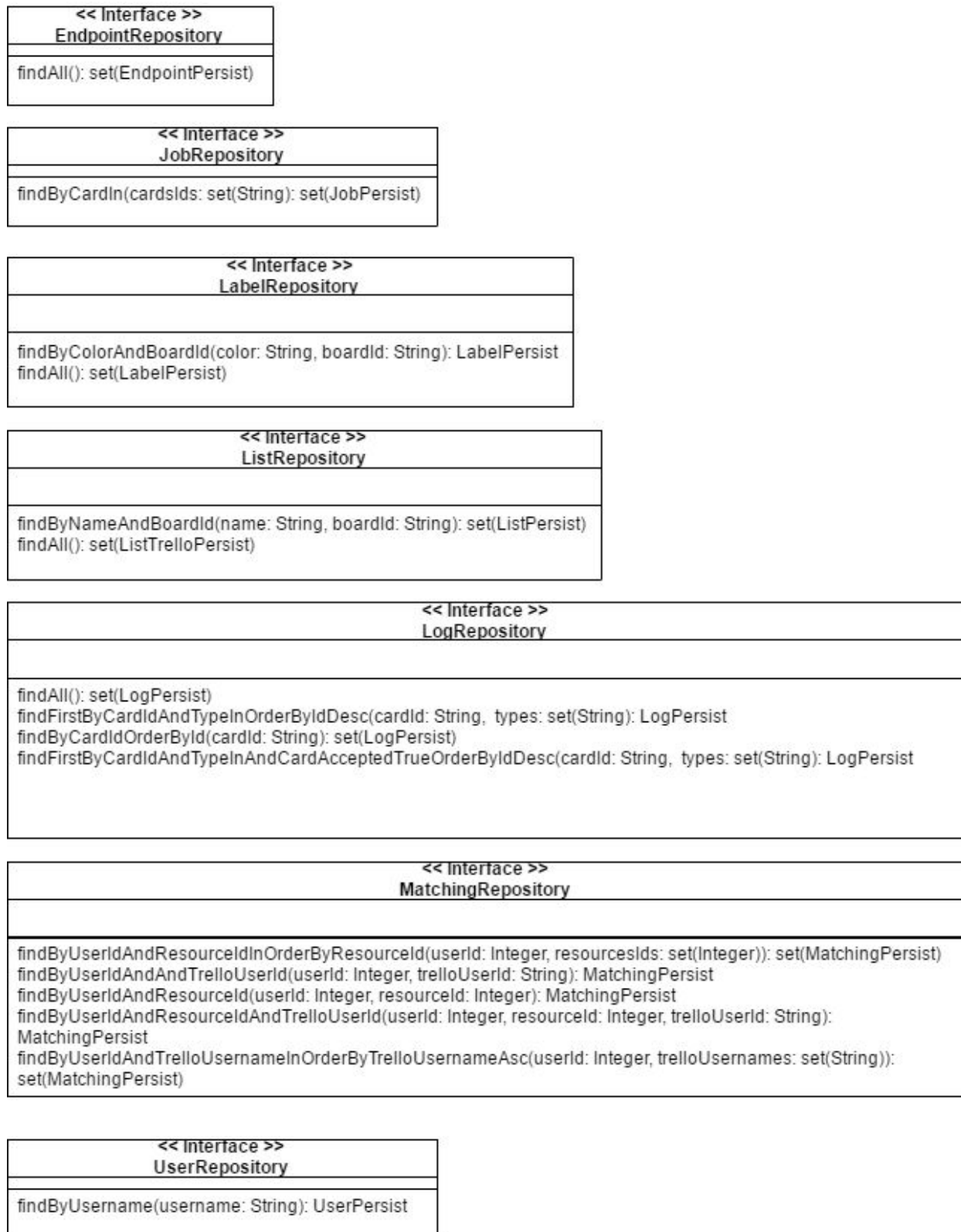


Fig 28. Diagrama classes interfície d'accés a dades (Font: elaboració pròpia)

La manipulació de les dades de persistència es fa a través de les classes del diagrama anterior. Aquestes classes són interfícies conegudes com a repositoris que hereten de CrudRepository. Spring Data JPA ofereix mètodes per guardar objectes o obtenir-ne proporcionant-ne la id. A més a més, permet crear *queries* a partir de *keywords* i seguint una sintaxi determinada ^[16]. Totes les funcions que apareixen en aquest diagrama són d'aquest tipus.

Domain Controller

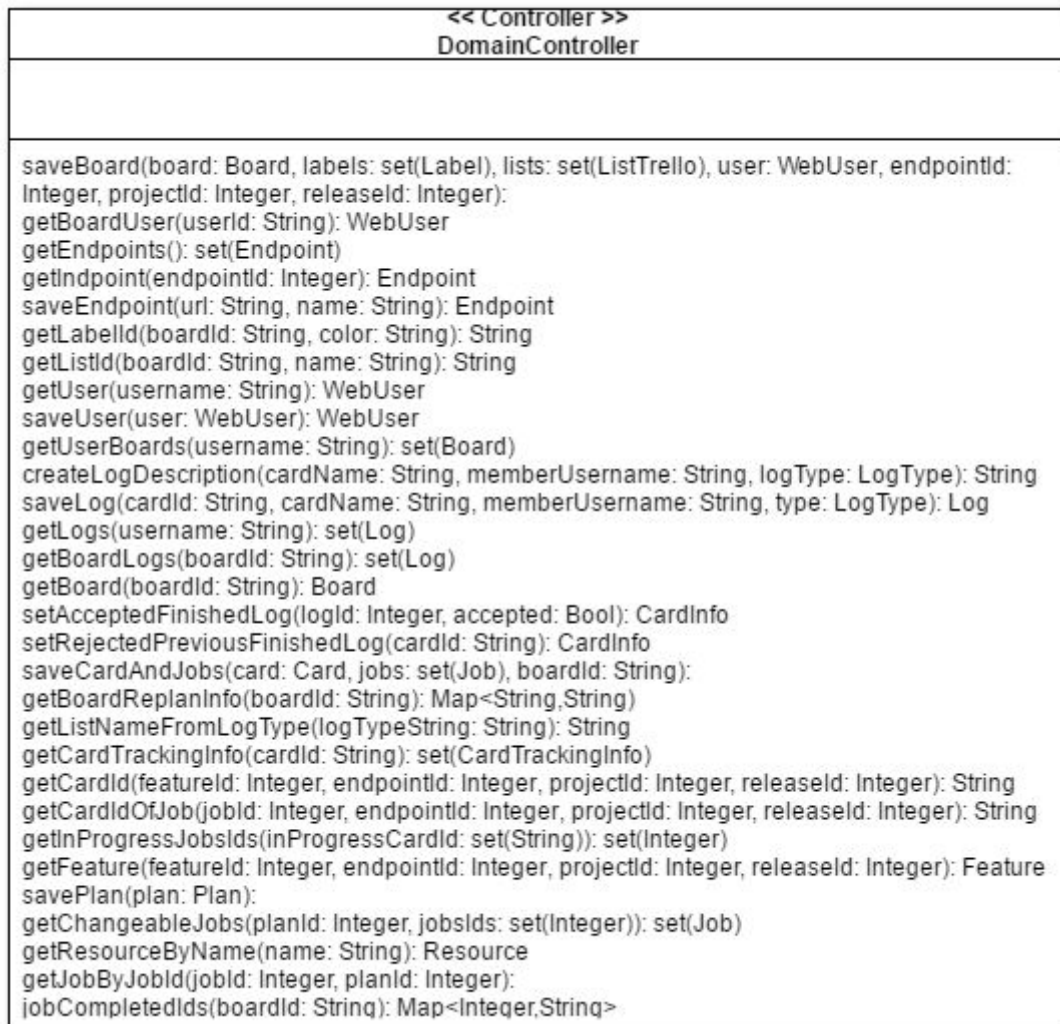


Fig 29. Diagrama classe Domain controller (Font: elaboració pròpia)

DomainController permet als REST controllers obtenir informació de persistència i guardar-ne. En el cas d'obtenir informació de persistència, utilitza els mètodes dels repositoris per aconseguir els objectes JPA per a després mapejar-los a DTOs i retornar-los. En el cas de tractar-se d'una operació que té com a objectiu guardar informació a persistència, mapeja els DTOs propis de la capa de serveis a objectes JPA del Domain Model i utilitza les funcions dels repositoris per guardar-los.

Relacions entre components domain layer

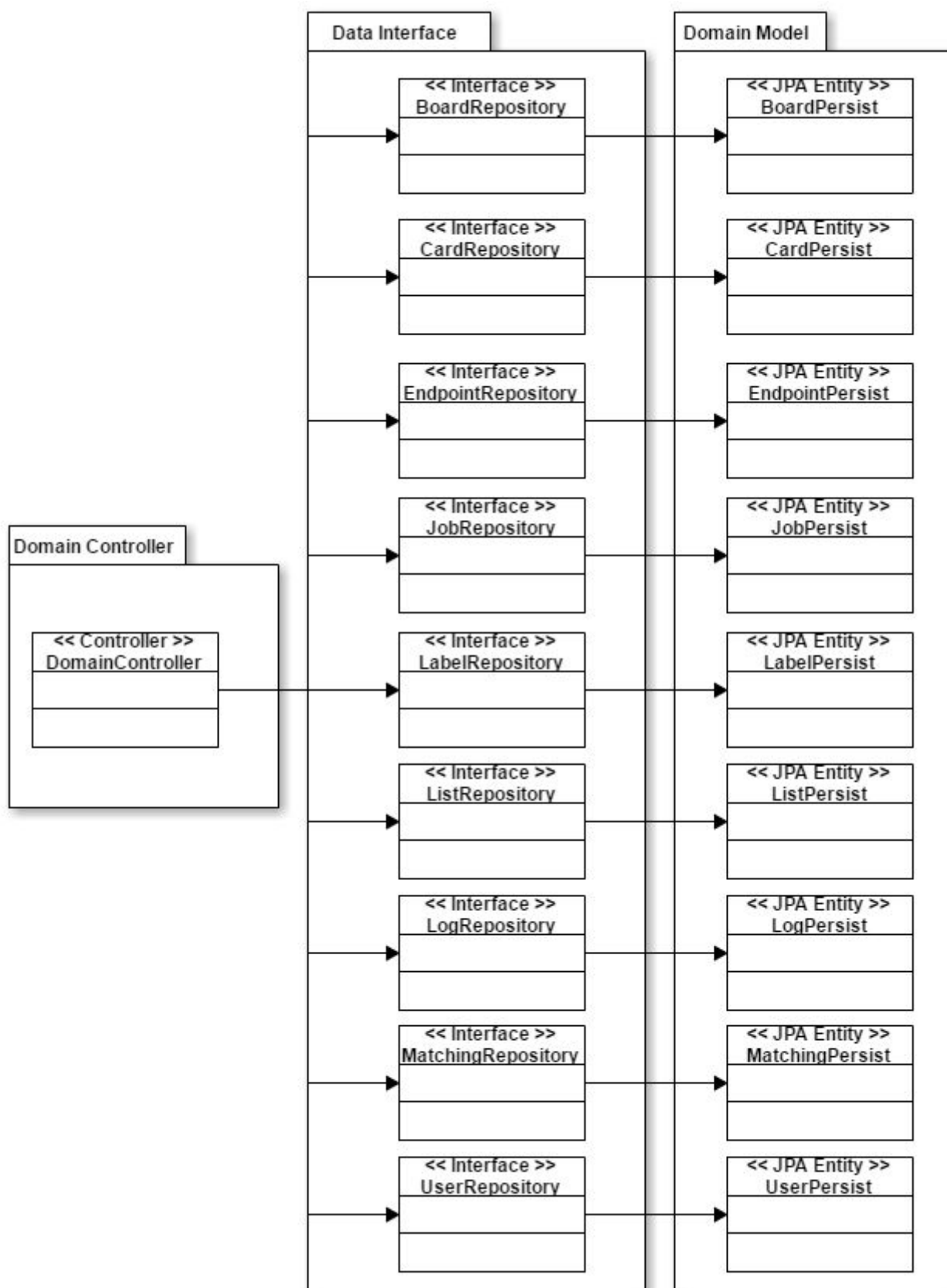


Fig 30. Diagrama interacció entre classes de la capa de domini (Font: elaboració pròpia)

10.2.2.4 Diagrama interacció entre capes de servidor

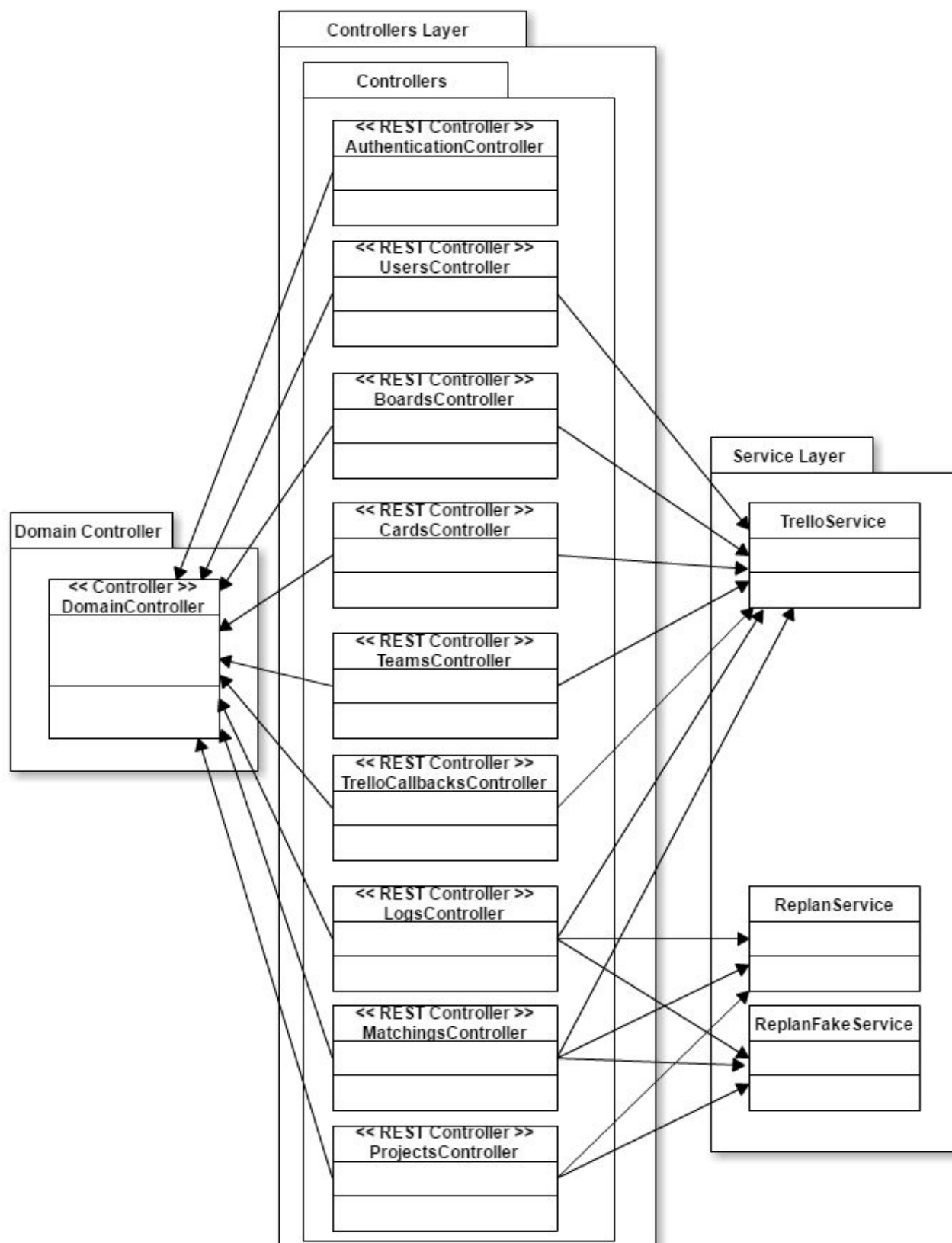


Fig 31. Diagrama interacció classes diferents capes del back-end (Font: elaboració pròpia)

10.2.2.5 Classes auxiliars

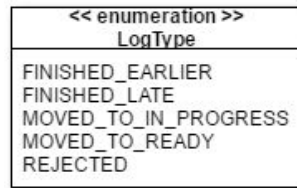


Fig 32. Diagrama classe enumeració (Font: elaboració pròpia)

Enumeració utilitzada per definir els tipus de logs. Els diferents tipus de logs són els següents:

- **FINISHED_EARLIER:** els logs d'aquest tipus es creen quan un usuari de Trello mou una card a la llista *Done* (marcant-la així com finalitzada) **abans** de la data d'acabament definida a la card.
- **FINISHED_LATE:** els logs d'aquest tipus es creen quan un usuari de Trello mou una card a la llista *Done* (marcant-la així com finalitzada) **després** de la data d'acabament definida a la card.
- **MOVED_TO_IN_PROGRESS:** els logs d'aquest tipus es creen quan un usuari de Trello mou una card a la llista *In Progress*.
- **MOVED_TO_READY:** els logs d'aquest tipus es creen quan un usuari de Trello mou una card a la llista *Ready*.
- **REJECTED:** els logs d'aquest tipus es creen quan el cap de projecte refusa una card des de la web.

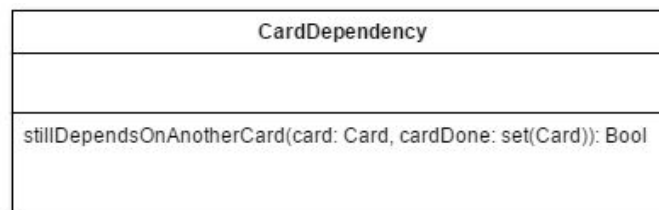


Fig 33. Diagrama classe auxiliar d'operacions (Font: elaboració pròpia)

Classe creada per reutilitzar una funció utilitzada a les classes **TrelloCallbacksController** i **LogsController**. Aquesta funció retorna si una card depèn actualment d'una altra card i és de vital importància a l'hora de calcular la següent card d'un membre.

11. Implementació

En aquesta secció es comentaran les eines i llenguatges que s'han utilitzat en la implementació del projecte.

El projecte s'ha implementat utilitzant l'IDE IntelliJ Ultimate 2017, que a part d'oferir suport per Java l'edició ultimate també ofereix suport al desenvolupament web. A IntelliJ s'ha creat un projecte Gradle que inclou les dependències de diferents mòduls que s'ha fet servir. Aquests mòduls són els corresponents a: Spring Boot Actuator, Spring Boot Web, Spring Boot Data JPA, Hibernate i SpringFox Swagger. SpringFox Swagger, que és l'únic del qual no s'havia parlat abans en el que va de document, permet generar l'especificació de l'API de manera automàtica.

Com ja s'ha mencionat al llarg del document, la part de back-end s'ha implementat amb el framework de Java Spring Boot i per tant, el llenguatge utilitzat en aquesta part ha estat Java.

Com també s'ha mencionat, la part de front-end s'ha implementat amb els *frameworks* AngularJs i Bootstrap. En aquesta part s'han utilitzat els llenguatges típics dels entorns web HTML5, CSS3 i JavaScript. Addicionalment s'han incorporat les llibreries també d'AngularJs angular-cookies i angular-routes així com la llibreria snackbar.

Per tal de garantir la disponibilitat del codi i facilitar la recuperació en cas de fallades, s'ha utilitzat l'eina de control de versions Git i el gestor de repositoris en xarxa GitHub. GitHub també ha estat necessari per poder desplegar l'aplicació remotament a Heroku. Per fer operacions sobre els repositoris i desplegar a Heroku s'ha utilitzat l'eina Git Bash i s'ha instal·lat HerokuCLI.

En el repositori¹, dins de *src/main/* es troba el codi propi de la implementació del Plugin dividit en 2 carpetes que contenen les parts de front-end i back-end. A continuació es detallarà el contingut de les carpetes:

- Java: conté la part de back-end, que es troba organitzada de la següent manera:
 - *Controllers*: conté les classes Java corresponents als REST Controllers.
 - *Domain_controllers*: conté la classe Java corresponent al controlador de domini.
 - *Dtos*: conté les classes Java utilitzades que s'utilitzen com a DTO en la comunicació per xarxa.
 - *Operation_classes*: conté la classe Java utilitzada per calcular les dependències actuals d'una targeta.
 - *Persistence*: conté les classes Java relacionades amb l'accés a les dades. Aquestes classes es troben repartides en quatre subcarpetes:

¹ Repositori disponible a <https://github.com/MiquelXamani/plugin-trello-replan>

- Models: conté les classes JPA.
- Repositories: conté les classes interfície per accedir a les dades.
- Fake_models: conté les classes JPA utilitzades en la simulació de la funcionalitat de replanificar.
- Fake_repositories: conté les classes interfície per accedir a les dades utilitzades en la simulació de la funcionalitat de replanificar.
- Plan_examples: conté els fitxers JSON de les planificacions que s'utilitzen en la funcionalitat de replanificar.
- Services: contés les classes Java que s'encarreguen de realitzar peticions als serveis externs i la classe que simula el comportament de la funcionalitat del Replan de replanificar.
- A més a més, conté les classes per inicialitzar el servidor i la base de dades, un adaptador per redirigir les peticions de contingut propi de front-end al front-end, la classe enumeration i la classe que permet generar l'especificació de l'API.
- Webapp: conté la part de front-end:
 - Fonts: conté els fitxers de fonts utilitzats per Bootstrap.
 - Css: conté els fitxers d'estil CSS.
 - Js: conté els fitxers JavaScript utilitzats (controladors, app.js, serveis...).
 - A més a més, conté els fitxers HTML utilitzats en les vistes i l'index.html.

A part del plugin, també ha estat necessari desplegar els components de l'eina Replan Replan Controller i Replan Optimizer. Per fer-ho, s'ha descarregat una còpia del repositori que els contenia i s'han desplegat per separat com a diferents aplicacions a Heroku. El codi del Replan Controller, implementat en Ruby on Rails, ha estat modificat per canviar l'endpoint del Replan Optimizer per la versió a Heroku i per modificar el fitxer *seeds*.

Per últim i no per això menys important, s'ha utilitzat Postman per testejar les APIs utilitzades en algun punt del projecte (Trello, Replan Controller, Replan Optimizer, Slack i la del propi plugin).

12. Testing

En aquesta secció s'explicarà com s'ha testejat i es veuran exemples de les operacions més importants que duu a terme el plugin.

12.1. Creació del tauler

Perquè el plugin creï un tauler d'una planificació necessita una planificació, un equip de Trello amb el qual associar el tauler, associacions entre els recursos de la planificació i els membres de l'equip de Trello i un nom per anomenar el tauler. Tota aquesta informació l'obté el plugin a través del seu front-end durant el procés de carregar una planificació a Trello.

Per comprovar que el plugin crea correctament els taulers a partir de les planificacions de la forma descrita en l'apartat 8. *Representació la planificació a Trello* s'han creat diferents releases dins de l'entorn del Replan.

La següent figura representa una planificació feta pel Replan d'una release que comença l'1 de juny (es pot veure en format JSON a l'*Annex D*):

Dia / recurs	1	2	3	4	5	6	7	8
Sergi	New channel							
Pere	Fix user display	Email reply				Enrollment refactoring		
Miquel	Fix auto upload	Profile picture						
Josep	New login							

Fig 34. Exemple planificació Replan (Font: elaboració pròpia)

En aquesta figura es pot veure per cada recurs quines tasques té assignades i la seva duració. Les línies que uneixen unes tasques amb altres representen dependència. Aquest exemple ha estat triat perquè té una característica interessant: una tasca depèn de la finalització d'unes altres dues per a ser finalitzada.

Aquest és el tauler resultant:

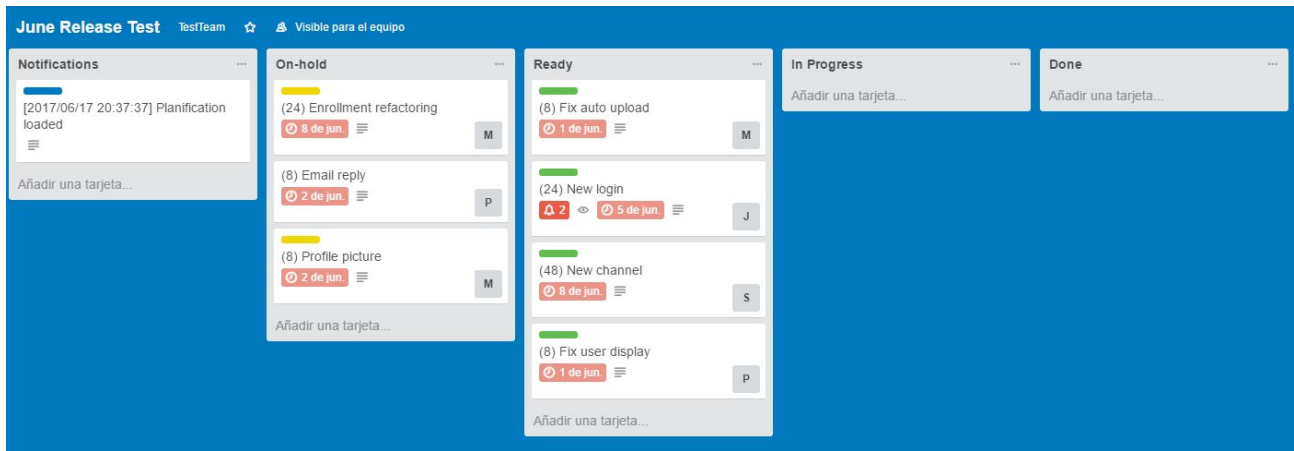


Fig 35. Captura planificació carregada a Trello (Font: elaboració pròpia)

Com es pot veure a la imatge, s'ha creat un tauler amb les llistes per defecte i amb les targetes corresponents a les tasques de la planificació. Com es pot veure, la llista **Ready** conté les targetes que ja es poden començar a fer mentre que les que encara no es poden començar a fer estan a la llista **On-hold**, ja sigui perquè depenen de la finalització d'una altra o perquè qui les ha de fer té assignada una que s'ha de realitzar abans. A més a més, s'han afegit les etiquetes grogues a aquelles targetes que depenen d'altres com és el cas de *Profile picture* i *Enrollment refactoring*, tal com hem vist a la figura anterior a aquesta.

12.2. Reaccionar quan es mou una targeta de llista

Per poder realitzar un seguiment de l'estat de les tasques a Trello és important saber quan una tasca canvia d'estat, és a dir, quan ha estat moguda de llista, qui ha realitzat aquest canvi i en quin estat es troba actualment.

Per tal de poder fer aquest seguiment, el plugin utilitza l'API de Trello per crear webhooks. Els webhooks permeten que el plugin pugui ser informat cada vegada que un membre realitza una acció sobre aquesta targeta. En el nostre cas, el que ens interessa és saber quan una targeta ha estat moguda d'una llista a una altra per tal d'enregistrar-ho i reaccionar a aquest moviment si escau.

Quan una targeta es mou a una altra llista, el plugin crea un registre (el que li hem dit **Log** en l'especificació). Aquests registres poden ser consultats a la web permetent d'aquesta manera que el cap de projecte pugui fer un seguiment més complet del transcurs de la release i fins i tot actuar des de la web acceptant o rebutjant una targeta que hagi estat marcada com finalitzada a Trello.

A part d'enregistrar aquests canvis, el plugin també realitza altres accions que repercuteixen sobre el tauler de Trello. Quan una targeta es mou a la llista Done, el plugin li posa una etiqueta una lila, elimina les etiquetes grogues d'aquelles targetes que depenien en aquell moment exclusivament de la targeta moguda i calcula la next card tant dels membres assignats a la card moguda com la dels membres assignats a una de les

targetes que depenien d'ella, movent aquestes next cards a la llista Ready i afegint-los una etiqueta verda a cadascuna.

També, per tal d'impedir un ús incorrecte del tauler, quan es detecta que s'ha fet un moviment erroni d'una targeta no enregistra l'acció i retorna la targeta a la llista d'on procedia.

Per testejar tant l'enregistrament com les reaccions, s'han creat diferents taulers de diferents planificacions sobre els quals s'han realitzat diferents moviments de forma manual simulant el transcurs normal d'una planificació. D'aquesta manera s'ha comprovat que tant el càlcul de les dependències com de la next card es realitzen correctament. També s'han realitzat moviments invàlids per tal d'assegurar que el plugin els desfà.

A continuació es mostra el tauler utilitzat en l'apartat anterior una vegada els desenvolupadors han començat a treballar en aquesta release:

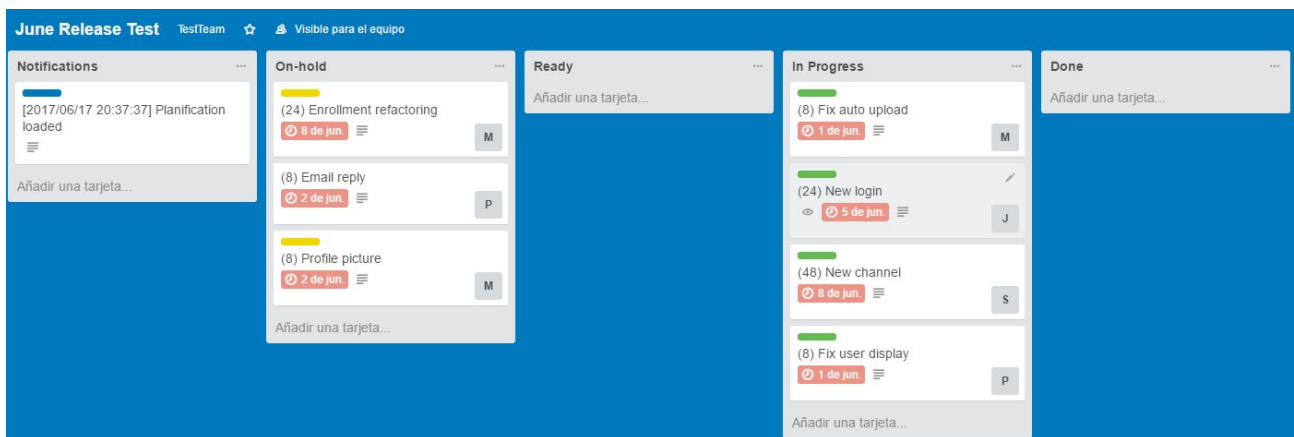


Fig 36. Targetes mogudes a la llista In Progress (Font: elaboració pròpia)

Com es pot veure a la imatge, s'han mogut les targetes *New channel*, *Fix user display*, *Fix auto upload* i *New login* a la llista *In Progress*.

I la pàgina on es mostra l'activitat del tauler mostra el següent:





Board Pending Actions				
Select a board				
June Release Test				
New logs				
Date	Board	Event	Description	Actions
17/06/2017 20:39:43	June Release Test	MOVEMENT	(8) Fix auto upload moved to in progress by josep248	
17/06/2017 20:39:44	June Release Test	MOVEMENT	(24) New login moved to in progress by josep248	
17/06/2017 20:39:47	June Release Test	MOVEMENT	(48) New channel moved to in progress by josep248	
17/06/2017 20:39:48	June Release Test	MOVEMENT	(8) Fix user display moved to in progress by josep248	

Fig 37. Captura pàgina home (Board pending actions) després d'haver mogut targetes a la llista In Progress (Font: elaboració pròpia)

Com es pot veure, aquesta pàgina permet visualitzar aquests moviments que han tingut lloc a Trello. En tractar-se d'un test, és un únic usuari el que ha mogut les targetes però en un cas real seria més habitual que cadascú es mogués les targetes que té assignades.

Seguint amb la planificació, s'ha mogut a la llista Done les targetes *Fix auto upload* i *Fix user display*:

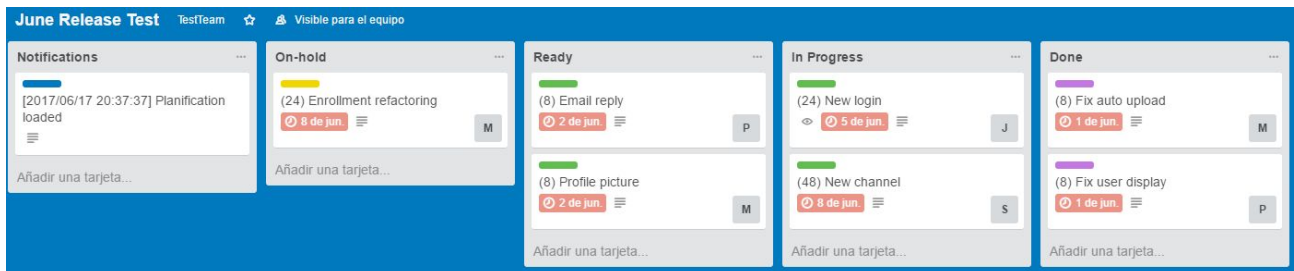


Fig 38. Targetes mogudes a la llista Done (Font: elaboració pròpia)

Les targetes mogudes a la llista *Done* tenen ara una etiqueta lila. Les targetes *Email Reply* i *Profile picture* han estat mogudes automàticament pel plugin a la llista *Ready*, cadascuna per diferents raons: *Email Reply* perquè és la next card d'en Pere i *Profile picture* perquè la targeta de la qual depenia (*Fix user display*) ha estat finalitzada i a més a més és la next card de Miquel.

Select a board				
June Release Test				
New logs				
Date	Board	Event	Description	Actions
17/06/2017 20:39:43	June Release Test	MOVEMENT	(8) Fix auto upload moved to in progress by josep248	
17/06/2017 20:39:44	June Release Test	MOVEMENT	(24) New login moved to in progress by josep248	
17/06/2017 20:39:47	June Release Test	MOVEMENT	(48) New channel moved to in progress by josep248	
17/06/2017 20:39:48	June Release Test	MOVEMENT	(8) Fix user display moved to in progress by josep248	
17/06/2017 20:51:08	June Release Test	FINISHED LATE	(8) Fix auto upload marked as finished by josep248 later than expected	
17/06/2017 20:51:15	June Release Test	FINISHED LATE	(8) Fix user display marked as finished by josep248 later than expected	
17/06/2017 20:51:16	June Release Test	MOVEMENT	(8) Email reply moved to ready by Project Leader	
17/06/2017 20:51:16	June Release Test	MOVEMENT	(8) Profile picture moved to ready by Project Leader	

Fig 39. Captura pàgina home (Board pending actions) després d'haver mogut targetes a la llista Done (Font: elaboració pròpia)

Com es pot veure en aquesta captura, el plugin ha enregistrat que s'han mogut les targetes a la llista *Done* i que el mateix plugin (sota el nom de project leader) ha mogut les targetes corresponents a la llista *Ready*. A més a més, s'ha detectat que aquestes targetes s'ha finalitzat més tard de la data prevista en la planificació i tenen accions pendents (acceptar i rebutjar).

12.3. Actualització del tauler

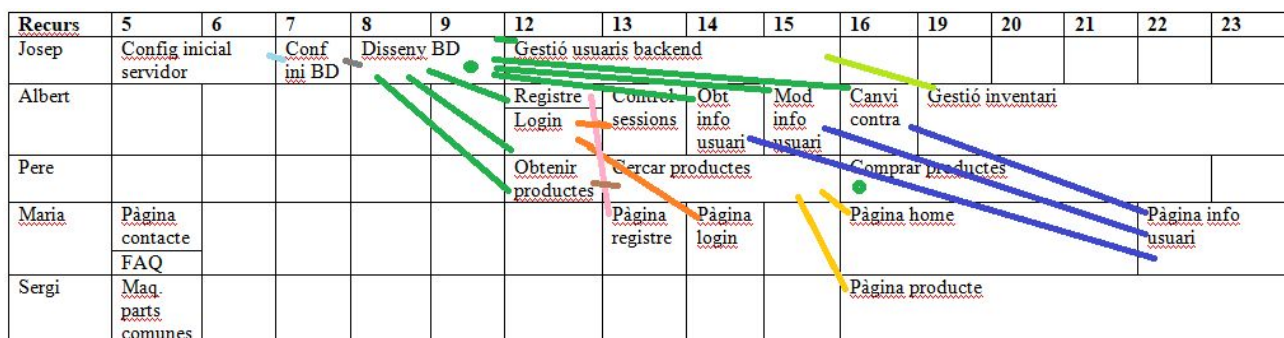
Quan el cap de projecte decideix replanificar una release, el plugin obté les dades que necessita el Replan per fer una replanificació. Una part de les dades les obté el plugin gràcies al fet que han estat guardades a persistència en el moment de la creació del tauler i associades a aquest. Aquestes dades són l'identificador del projecte, l'identificador de la release i l'url on fer la crida. L'altra part és la informació relativa a l'estat de la planificació perquè el Replan tingui en compte l'estat per tal de fer la replanificació. Aquestes dades són les tasques (concretament els identificadors dels jobs) ja acabades juntament amb la data de finalització a Trello i les tasques en progrés.

Tot seguit es realitzaria una crida a l'API del Replan passant-li aquesta informació. La resposta d'aquesta crida seria una nova planificació de la mateixa release que pot haver introduït canvis en la planificació. A partir d'aquesta resposta, el plugin actualitzaria el contingut del tauler de Trello corresponent.

Malauradament, la funcionalitat del Replan de replanificar no ha estat finalitzada i s'ha optat per simular la seva resposta. Per fer-ho, s'ha creat una classe que simula aquesta operació del Replan donant respostes fixades per una planificació en concret i situacions concretes.

Per comprovar el correcte funcionament de l'actualització del tauler seguint els canvis introduïts en la replanificació s'ha creat una planificació d'una release. Aquesta release comença el 5 de juny i acaba el 23 del mateix mes.

La següent figura representa aquesta planificació que s'ha fet manualment seguint el format de les planificacions que genera l'eina Replan (es pot veure la planificació en JSON a l'Annex E):



Recurs	5	6	7	8	9	12	13	14	15	16	19	20	21	22	23
Josep	Config inicial servidor		Conf ini BD	Disseny BD		Gestió usuaris backend									
Albert						Registre Login	Control sessions	Obt info usuari	Mod info usuari	Canvi contra	Gestió inventari				
Pere						Obtenir productes	Buscar productes			Comprar productes					
Maria	Pàgina contacte						Pàgina registre	Pàgina login		Pàgina home				Pàgina info usuari	
Sergi	Maq. parts comunes									Pàgina producte					

Fig 40. Planificació utilitzada per replanificar (Font: elaboració pròpia)

La planificació representa una primera release d'un e-commerce de llibres. En aquesta release intervenen recursos especialitzats en diferents àrees, Josep s'encarregaria de la part d'administració de bases de dades, Albert i Pere del back-end i Maria i Sergi del front-end.

S'ha optat per una planificació més complexa perquè s'assembli més a un cas real i poder comprovar que l'actualització del tauler funciona en casos amb múltiples dependències. També cal saber que les jornades laborals són de 8 hores (de 9:00 a 17:00) i que les cel·les on hi ha més d'una tasca vol dir que duren la meitat de la jornada cadascuna, és a dir 4 hores.

Aquest és el tauler resultant del procés de carregar la planificació a Trello:

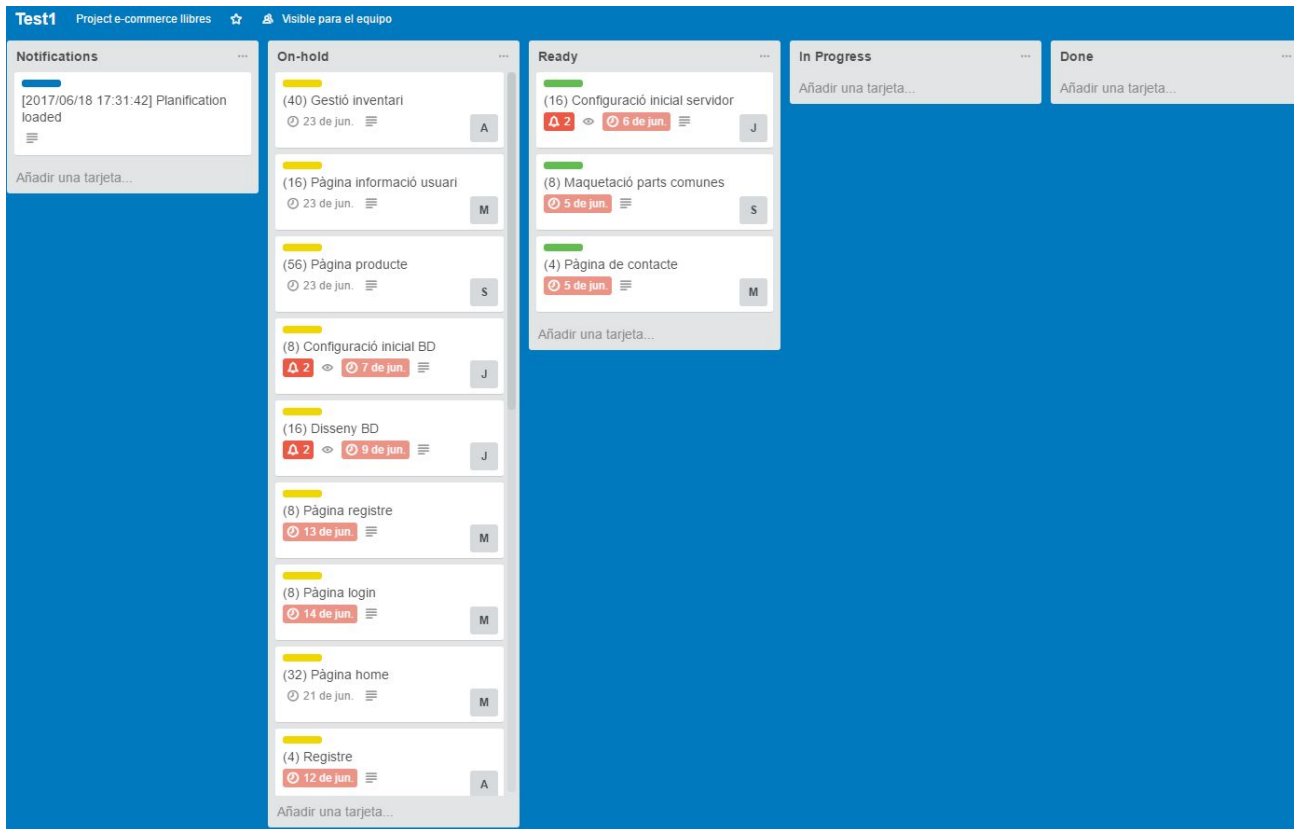


Fig 41. Planificació utilitzada per replanificar a Trello (Font: elaboració pròpia)

A continuació es mostraran exemples d'escenaris on s'ha realitzat una replanificació:

Cas 1: desaparició de tasques

En aquest cas es va complir la planificació fins que la tasca Disseny BD finalitza 1 dia més tard del previst.

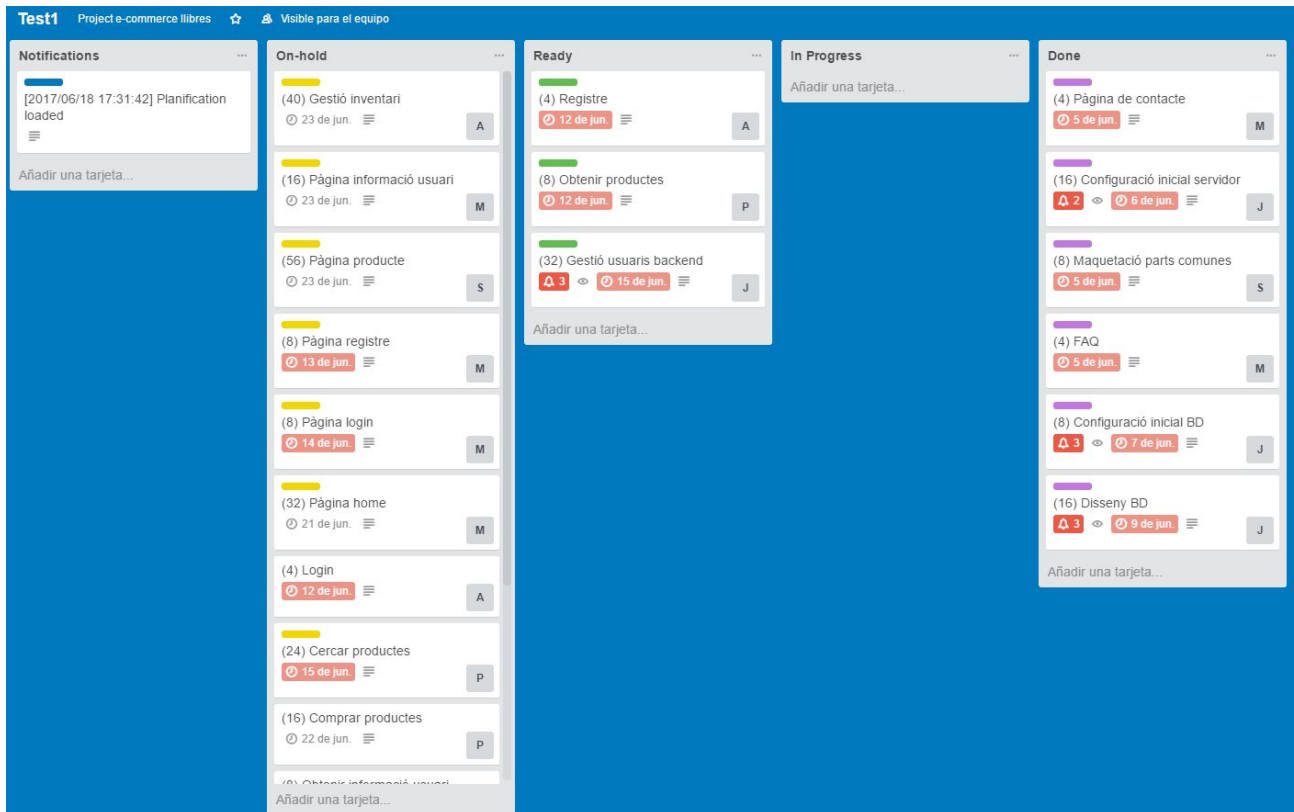


Fig 42. Tauler abans de replanificar (cas eliminació de targetes) (Font: elaboració pròpia)

Davant aquesta situació, el cap de projecte accepta la tasca com a finalitzada i prem per replanificar.

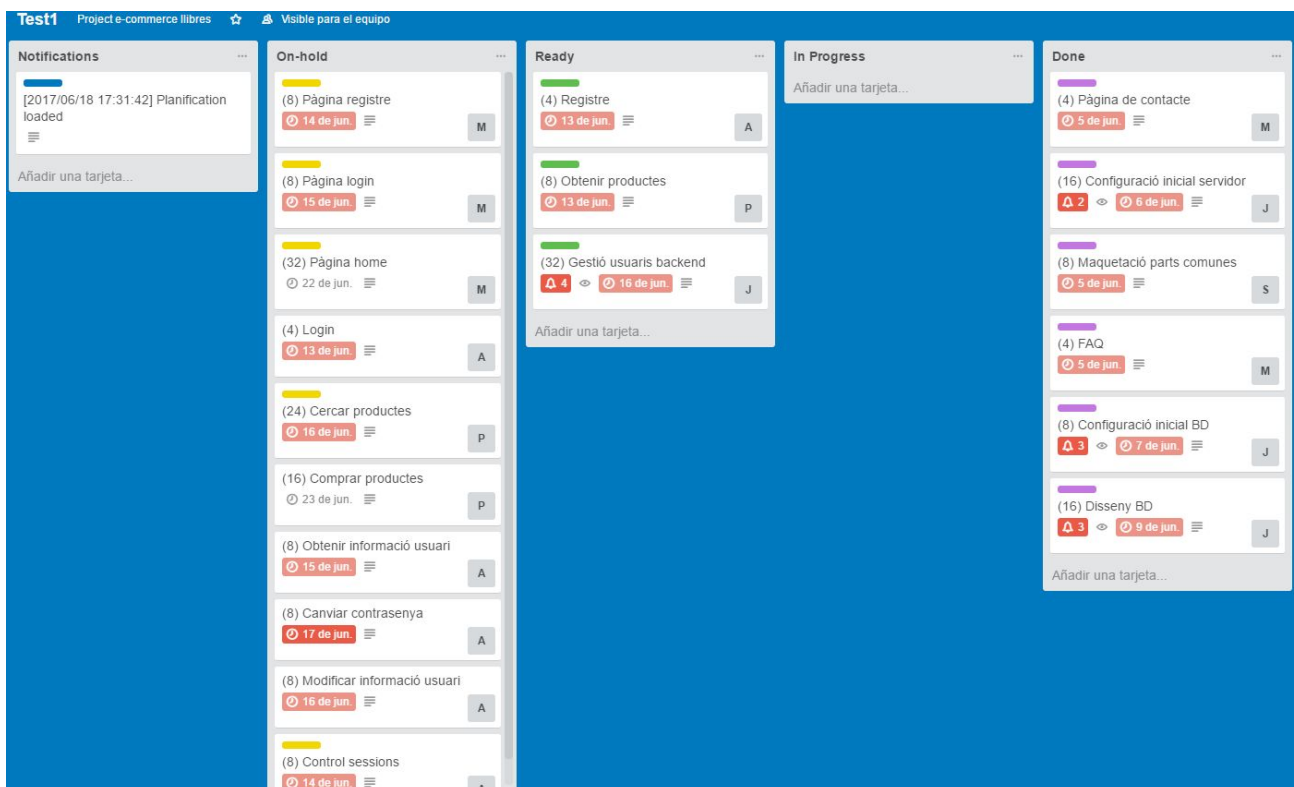


Fig 43. Tauler després de replanificar (cas eliminació de targetes) (Font: elaboració pròpia)

En la nova planificació s'ha endarrerit la resta de tasques un dia a excepció de les tasques que acabaven el mateix dia que la data límit de la release (*Gestió inventari*, *Pàgina informació usuari* i *Pàgina producte*), que han estat eliminades de la planificació i per tant, del tauler. Aquestes targetes que han estat eliminades són les 3 primeres que estaven en la llista *On-hold* abans de replanificar.

Cas 2: aparició de nova tasca

En aquest cas, es va complint la planificació fins al dia 20 de juny, que Sergi acaba *Pàgina del producte* 3 dies abans de la data fixada. El tauler en aquest moment és el següent:

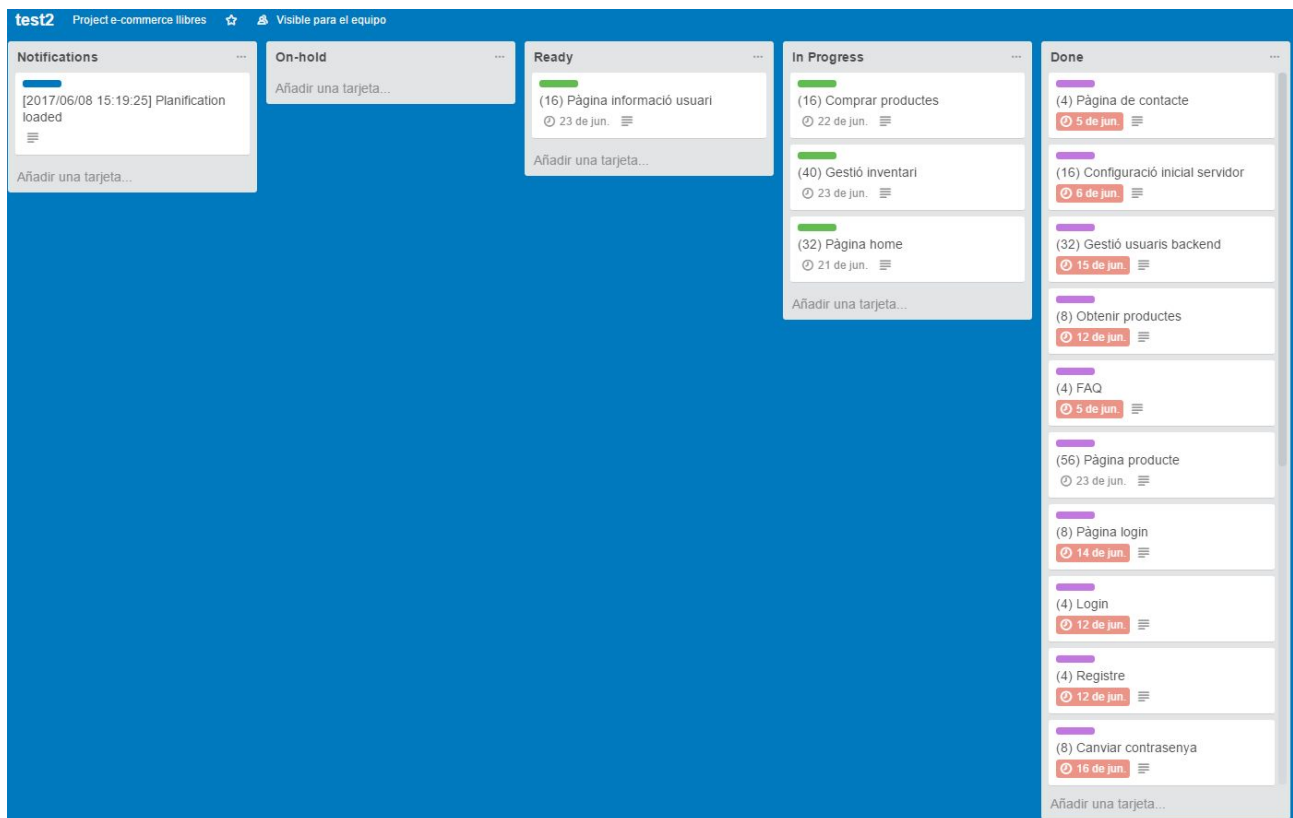


Fig 44. Tauler abans de replanificar (cas aparició de nova targeta) (Font: elaboració pròpia)

Tot seguit, el cap de projecte accepta com a finalitzada aquesta targeta i li dóna al botó de replanificar. Així es veu el tauler després de la replanificació.

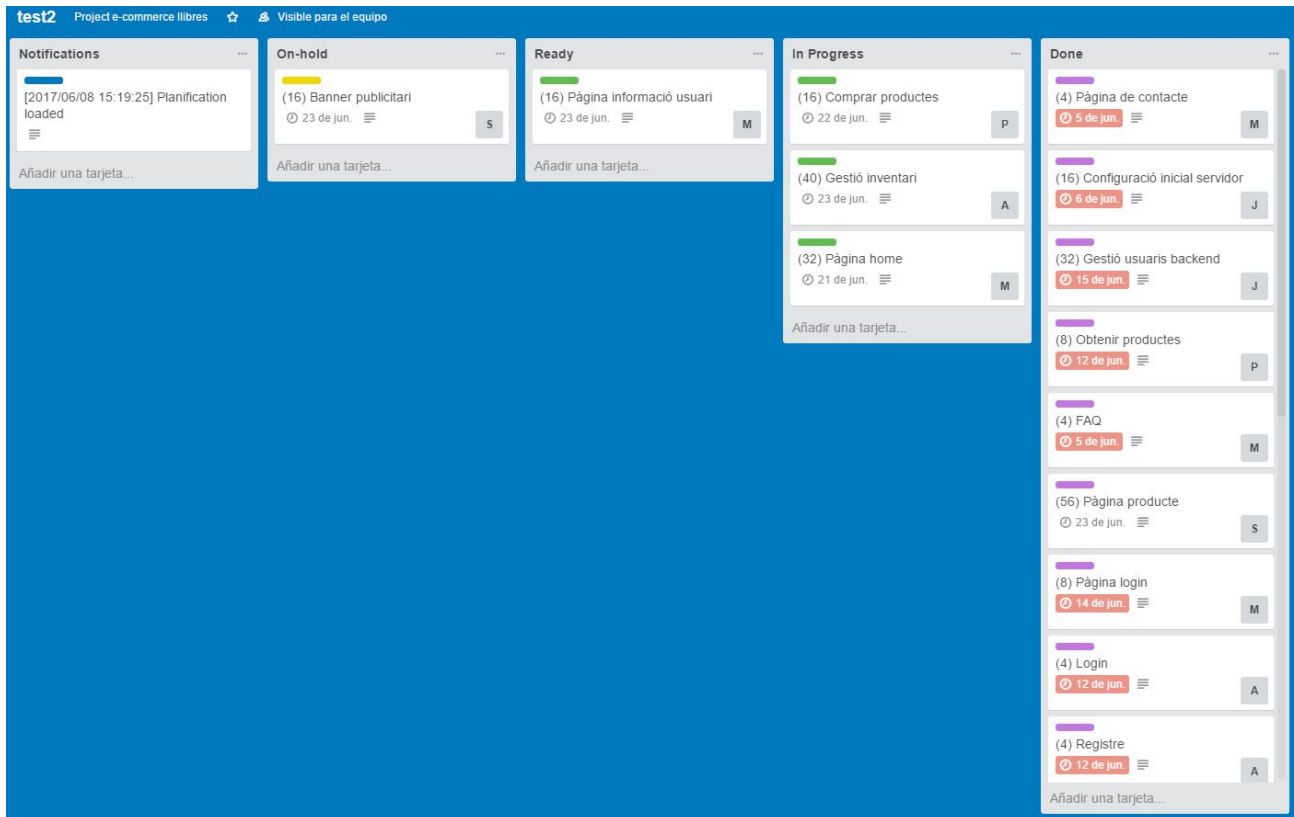


Fig 45. Tauler després de replanificar (cas aparició de nova targeta) (Font: elaboració pròpia)

En aquesta captura del tauler després de la replanificació es pot veure que ha aparegut una targeta nova (*Banner publicitari*) a la llista On-hold. Aquesta targeta correspon a una tasca que requereix habilitats de front-end, que formava part de la release i depèn de *Pàgina home* però que havia estat deixada a fora perquè no hi cabia. La targeta ha estat ara afegida a la planificació perquè *Pàgina producte* ha estat finalitzada i per tant en Sergi té 3 dies lliures, que és el que està previst que duri aquesta nova targeta segons l'esforç estimat. També està previst que *Pàgina home* acabi 2 dies abans del final de la release, tal com diu la planificació. Per tant, el Replan ha fixat la data de començament d'aquesta targeta pel dia 22 de juny (quan *Pàgina home* ha acabat) i la seva finalització el 23 del mateix mes, coincidint amb la data límit de la release.

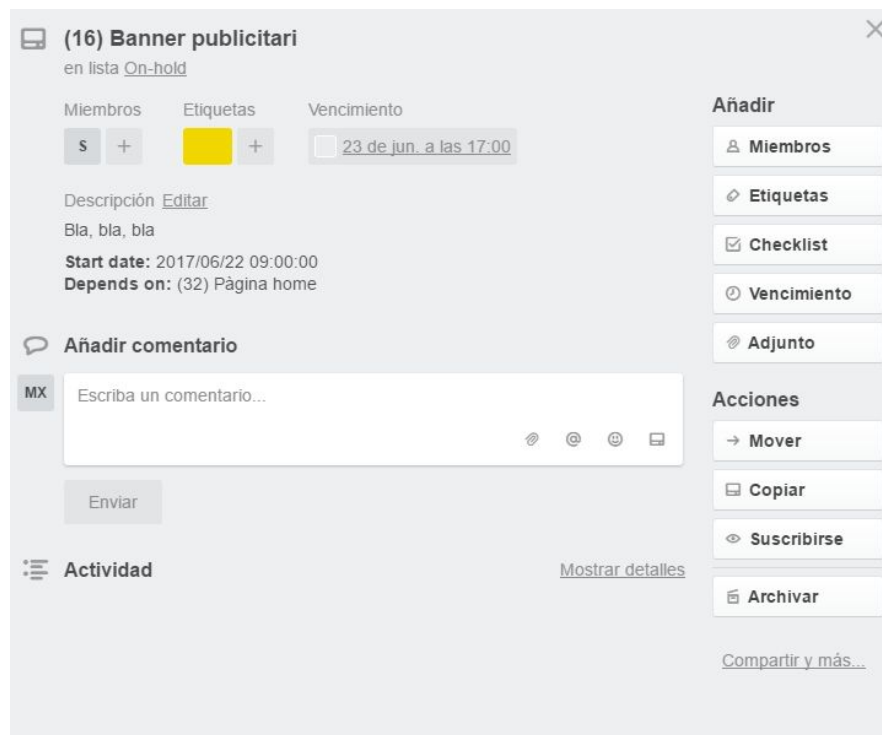


Fig 46. Detall de la targeta afegida per la replanificació (Font: elaboració pròpia)

La targeta es troba a la llista *On-hold* perquè depèn de la finalització de *Pàgina home*, una targeta que està actualment en progrés i per aquesta dependència se li ha afegit una etiqueta groga.

Cas 3: Canvi d'assignació

En aquest es va complint la planificació fins que Sergi acaba *Maquetació de parts comunes* a les 11:00, 6 hores abans del previst. La Maria per la seva banda continua treballant amb *Pàgina de contacte*.

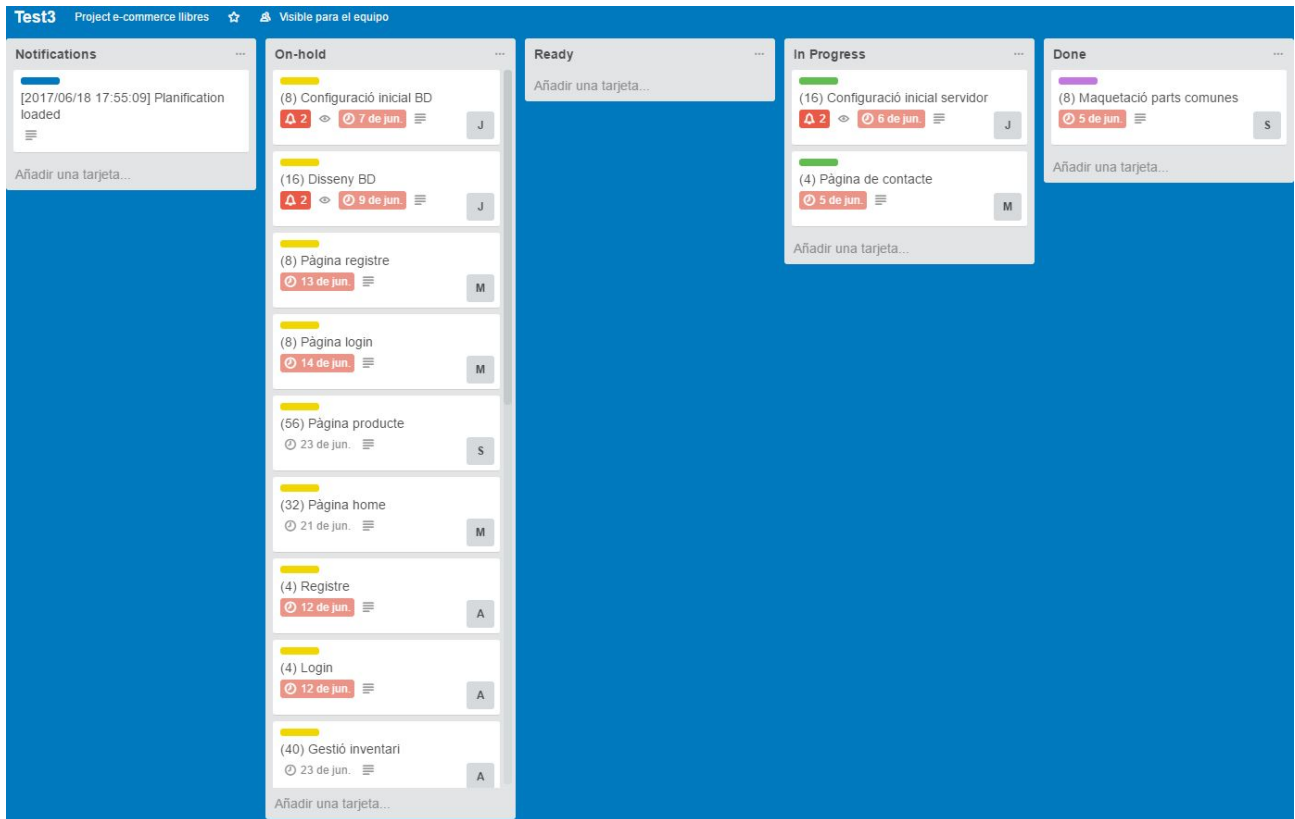


Fig 47. Tauler abans de replanificar (cas canvis d'assignació) (Font: elaboració pròpia)

El cap de projecte accepta la tasca com a finalitzada i prem el botó de replanificar.

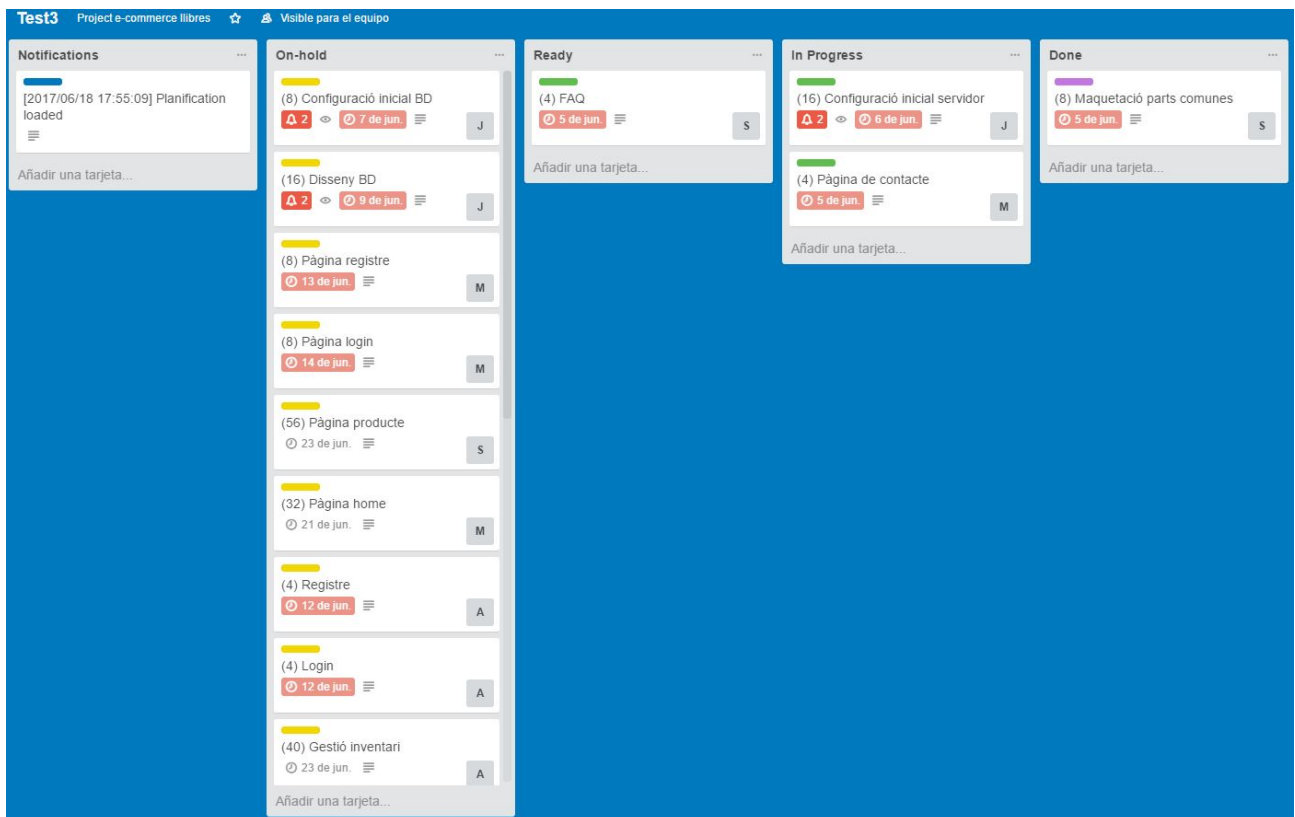
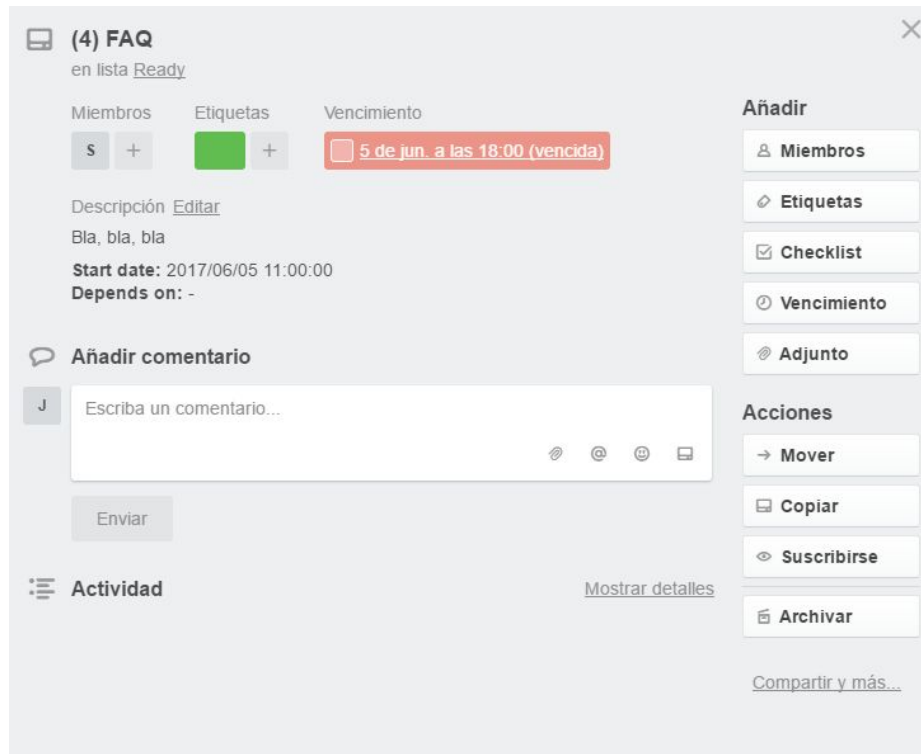


Fig 48. Tauler després de replanificar (cas canvis d'assignació) (Font: elaboració pròpia)



(4) FAQ
en lista [Ready](#)

Miembros + Etiquetas + Vencimiento

Descripción [Editar](#)
Bla, bla, bla
Start date: 2017/06/05 11:00:00
Depends on: -

Añadir comentario
J

Actividad [Mostrar detalles](#)

Añadir
[Miembros](#)
[Etiquetas](#)
[Checklist](#)
[Vencimiento](#)
[Adjunto](#)

Acciones
[→ Mover](#)
[Copiar](#)
[Suscribirse](#)
[Archivar](#)
[Compartir y más...](#)

Fig 49. Detall de la targeta del canvi d'assignació (Font: elaboració pròpia)

Com es pot veure en les dues captures anteriors, la tasca *FAQ* que estava assignada a Maria ara està assignada a Sergi i les dates de començament i acabament han estat actualitzades.

13. Desplegament

El **plugin** s'ha desplegat a Heroku (<https://glacial-anchorage-60164.herokuapp.com/>). El desplegament del plugin no ha estat una tasca que s'hagi deixat per la fase final sinó que des de la primera fase de construcció porta realitzant-se. El motiu d'aquest desplegament en etapes tan prematures ha estat per corregir els possibles errors generats pel desplegament a Heroku abans que el projecte guanyés en complexitat i fos més difícil corregir aquests possibles errors. A partir del primer desplegament, s'han anat realitzant-ne més de forma esporàdica, ja que es podia testejar en local fent que no fos necessari desplegar-ho en el núvol. No va ser fins al moment d'implementar la detecció de canvis en les targetes que es va tornar obligatori. La raó és que perquè Trello notifiqui al plugin de canvis en les tasques és necessari proveir un url vàlida en el moment de la creació dels webhooks que monitoren les targetes, ja que Trello fa 3 peticions *Head* a l'url facilitada per assegurar-se que és vàlida.

Tant el **Replan Controller** (<https://lit-savannah-17077.herokuapp.com/>) com el **Replan Optimizer** (<https://intense-bastion-31216.herokuapp.com/>) també estan desplegats a Heroku.

A la figura següent es mostra el desplegament final del sistema:

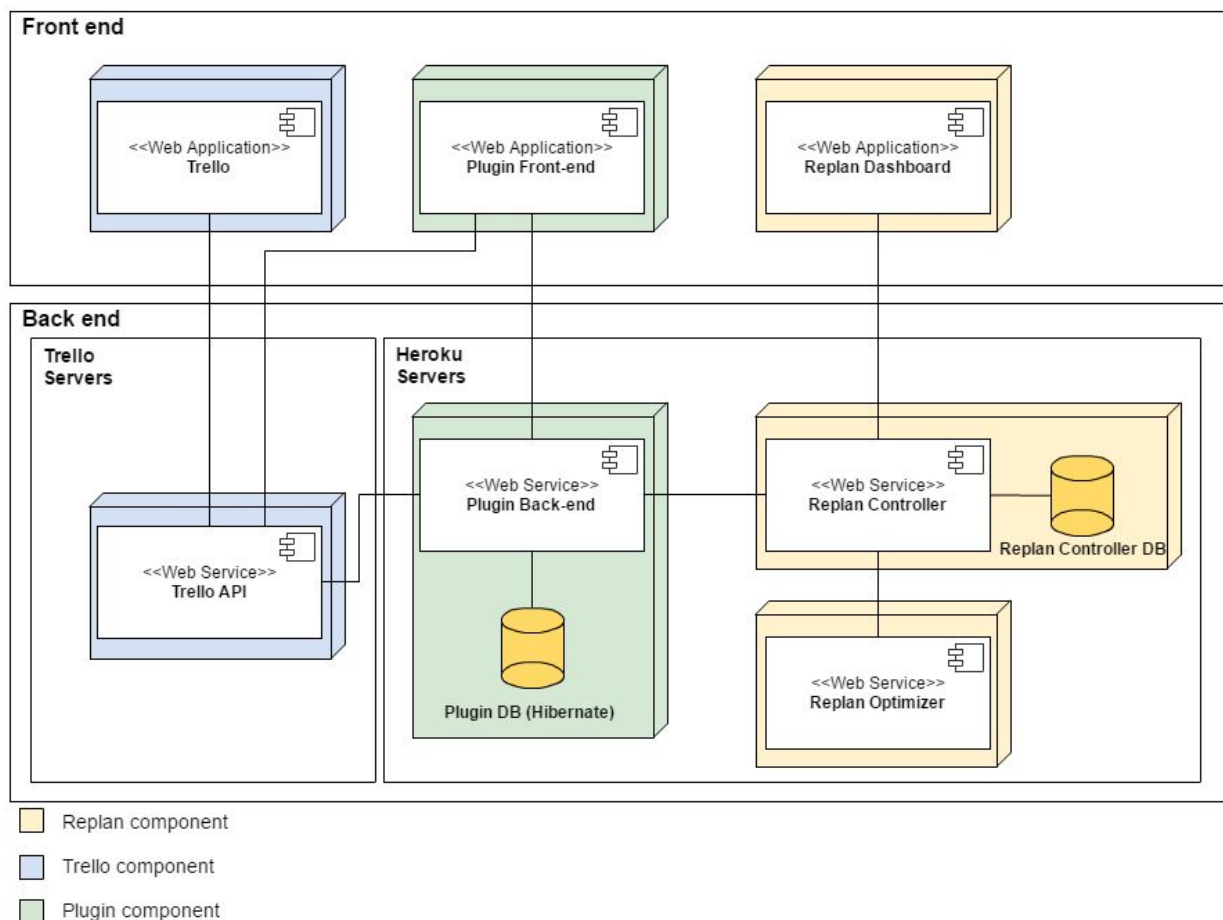


Fig 50. Diagrama de desplegament (Font: elaboració pròpia)

14. Sostenibilitat i compromís social

A continuació s'explica l'informe de sostenibilitat del projecte per a les dimensions ambiental, econòmica i social, les tres dimensions de la sostenibilitat.

	PPP	Vida útil	Riscs
Ambiental	7	17	0
Econòmic	8	15	0
Social	9	17	0
Rang de sostenibilitat	24/30	49/60	0
	73/90		

Taula 20. Matriu de sostenibilitat. (Font: elaboració pròpia)

14.1. Dimensió ambiental

La realització d'aquest projecte no necessita nous recursos materials, ja que es disposaven i s'utilitzaven abans del començament d'aquest. L'ordinador utilitzat i el *router* tenen un consum elèctric però com no s'utilitzen exclusivament per realitzar aquest projecte no suposen un increment significatiu en el consum.

S'utilitzaran eines i serveis ja existents, estalviant així temps i diners en desenvolupament. L'ús d'aquests serveis implica un consum d'energia per part dels proveïdors d'aquests serveis però com aquest projecte no és l'únic que els utilitza l'increment en el consum ocasionat per les activitats realitzades en el projecte és ínfim.

Es desplegarà el producte en servidors *cloud* estalviant així en infraestructura i costos en l'espai d'emmagatzematge de dades. D'altra banda, com més quantitat de dades i connexions rebí el producte, més consum per part dels servidors on es troba allotjat i per tant més perjudicial per al medi ambient.

És important recordar que el projecte és un producte *software* i per tant no té costos de fabricació ni generarà contaminació. Tampoc es preveu el desmantellament de cap recurs material en acabar el projecte.

14.2. Dimensió econòmica

Es defineixen correctament els costos materials i humans del projecte i s'han estimat els temps proporcionalment a la seva importància dins del projecte. Destacar també que durant el desenvolupament hi haurà una constant gestió dels riscos.

14.3. Dimensió social

Amb el resultat d'aquest projecte s'aconseguirà una eina que serà capaç d'enllaçar l'eina Replan amb una plataforma de suport al desenvolupament col·laboratiu. D'aquesta manera els caps de projecte disposaran d'una eina que permet **Continuos Software Release Planning** que facilitarà la seva feina. Els desenvolupadors hauran de modificar lleugerament el seu ús de la plataforma si el cap de projecte decideix utilitzar l'eina però el període d'adaptació serà insignificant.

15. Conclusions

15.1. Assoliment dels objectius

Els objectius presentats en un primer moment en la formulació del projecte s'han assolit de forma satisfactòria. S'ha escollit una de les plataformes de suport al desenvolupament col·laboratiu amb la qual fer una integració amb l'eina Replan. L'elecció de la plataforma, que finalment ha estat Trello, ha estat realitzada a partir de les conclusions obtingudes de l'estudi de les plataformes candidates.

Una vegada escollida la plataforma, s'ha desenvolupat un plugin que permet la integració de l'eina Replan amb la plataforma escollida. Amb la implementació de les funcionalitats especificades, s'ha obtingut una eina que permet als desenvolupadors treballar directament sobre les planificacions generades pel Replan a Trello, fet que permet que es pugui monitorar l'activitat d'aquests en la plataforma, assolint així un dels objectius. Gràcies a aquesta monitoratge, el cap de projecte pot realitzar un seguiment més acurat del transcurs de la release a través de l'aplicació web que també implementa el plugin. Mitjançant aquesta aplicació web, el cap de projecte pot acceptar i rebutjar tasques que han estat marcades com acabades a Trello pels desenvolupadors i el que és més important, el cap de projecte pot decidir que el Replan replanifiqui una release en curs.

Per tal que el Replan pugui generar una replanificació, necessita el feedback de l'activitat que realitzen els desenvolupadors a Trello, *feedback* que és proveït pel plugin en el moment de demanar la replanificació, assolint així el darrer dels objectius.

15.2. Coneixements aplicats i adquirits

Tal com es va explicar en l'elecció de les competències d'aquest projecte, durant el seu desenvolupament s'han aplicat diversos i nombrosos conceptes relacionats amb l'Enginyeria del Software. Per una banda s'han reafirmat i complementat els coneixements adquirits durant el grau com són l'especificació i anàlisi de requisits, l'anàlisi de sistemes, les premisses de disseny, els fluxos d'execució, l'ús d'APIs i la gestió de projectes.

D'altra banda, s'ha tingut l'oportunitat d'aprendre a treballar amb tecnologies amb les quals no es tenia experiència com Spring i els seus components (Spring Data JPA, Spring Boot Actuator...). També s'ha guanyat experiència en AngularJS i Bootstrap, tecnologies sobre les quals es tenia un coneixement molt bàsic abans de començar el projecte.

15.3. Treball futur

Encara que el TFG s'ha finalitzat, es pot seguir desenvolupant el projecte. L'actualització del tauler s'ha realitzat simulant la resposta que el Replan donaria en el cas de demanar una replanificació. Una vegada el Replan disposi de la funcionalitat de replanificar, el plugin utilitzarà aquesta funcionalitat, en comptes de simular-la. Per aconseguir-ho, és molt possible que s'hagi d'adaptar la informació a enviar d'acord amb una nova especificació d'aquesta operació de l'API del Replan Controller.

En aquest treball ens hem centrat en el moment de finalitzar les tasques, però també seria interessant informar els usuaris de les tasques que estan a punt de finalitzar i les tasques que ja han vençut via correu electrònic. Aquestes funcionalitats formaven part de l'abast inicial però finalment han quedat fora.

Com bé s'havia comentat en l'apartat 5.4. *Canvis en la planificació*, la funcionalitat *Consultar canvis d'estat d'una tasca* obre una porta a l'explotació de les dades. L'explotació d'aquestes dades permetria que el cap de projecte disposés de dades interessants com la diferència entre l'esforç estimat i el real, el rendiment dels desenvolupadors o fins i tot saber el nombre de tasques rebutjades per detectar si hi ha problemes a l'hora de definir els requisits. Aquesta explotació de les dades seria treball a realitzar en un futur.

Per últim, durant l'anàlisi i especificació de requisits es va estudiar les possibles circumstàncies que es podrien donar que podrien causar un canvi en la planificació actual, sent necessària una replanificació. En aquest treball ens hem centrat a detectar si hi ha retards en la finalització de les tasques o per contra si acaben abans de l'esperat, que són situacions que tenen lloc a Trello però en l'àmbit del Replan, també es poden donar altres circumstàncies que fan que sigui necessària una replanificació. Algunes d'aquestes circumstàncies serien:

- Un recurs deixa de formar part d'una *release* per diversos motius (està de baixa, ja no forma part de l'empresa o l'han exclòs de la *release* en qüestió). Aquesta situació requeriria canvis en la planificació per tal mitigar l'impacte d'aquesta baixa en la *release*.
- S'incorpora un nou recurs a una *release*, fet que requeriria una replanificació per tal de redistribuir la feina a fer entre els recursos que formen part de la *release*.

Per tant, una feina a fer seria que el Replan informés el plugin d'aquests canvis per tal que el cap de projecte fos informat i replanifiqués.

16. Referències

- [1] SUPERSEDE. *Supersede* [en línia]. [Consultada: 22 Febrer 2017]. Disponible a: < <https://www.supersede.eu/> >
- [2] Ameller, D., Cassarino, A., Elvassore, V., Farré, C., Franch, X., Valerio, D.. *Replan: A release planning tool* [en línia]. [Consultada: 22 Febrer 2017]. Disponible a: < <http://www.essi.upc.edu/dameller/publications/ameller2017-saner-demo.pdf> >
- [3] Ameller, D., Cassarino, A., Elvassore, V., Farré, C., Franch, X., Valerio, D.. *Replan: A release planning tool* [en línia]. [Consultada: 22 Febrer 2017]. Disponible a: <<http://www.essi.upc.edu/dameller/publications/ameller2017-saner-era.pdf> >
- [4] Atlassian. *JIRA Software: Supervisión de problemas y proyectos para equipos de software* | Atlassian [en línia]. [Consultada: 23 Febrer 2017]. Disponible a: < <https://es.atlassian.com/software/jira> >
- [5] Slack. *Slack: Where work happens* [en línia]. [Consultada: 23 Febrer 2017]. Disponible a: < <https://slack.com/> >
- [6] Slack. *Getting started with Slack apps* | Slack [en línia]. [Consultada: 23 Febrer 2017]. Disponible a: < <https://api.slack.com/slack-apps> >
- [7] Trello. *Trello* [en línia]. [Consultada: 23 Febrer 2017]. Disponible a: < <https://trello.com/> >
- [8] Trello. *Manténgase al tanto con los Power-Ups de Trello* [en línia]. [Consultada: 23 Febrer 2017]. Disponible a: < <https://trello.com/power-ups> >
- [9] Trello. *Trello Developers* [en línia]. [Consultada: 23 Febrer 2017]. Disponible a: < <https://developers.trello.com/> >
- [10] Proyectosagiles.org. *Desarrollo iterativo e incremental - Proyectos Ágiles* [en línia]. [Consultada: 26 Febrer 2017]. Disponible a: < <https://proyectosagiles.org/desarrollo-iterativo-incremental/> >
- [11] Page Personnel. *Page Personnel | Consultoría de Selección Especializada* [en línia]. [Consultada: 9 Març 2017]. Disponible a: < <http://www.pagepersonnel.es/> >
- [12] Michael Page. *Michael Page | Selección de Ejecutivos y Directivos* [en línia]. [Consultada: 9 Març 2017]. Disponible a: < <http://www.michaelpage.es/> >
- [13] Rodalies de Catalunya. *Servicio integrado ATM* [en línia]. [Consultada: 9 Març 2017]. Disponible a: < http://rodalies.gencat.cat/es/tarifes/servei_rodalia_barcelona/generals/servei_integrat_atm/#FW_bl oc_f49e49ab-1d79-11e4-8ddc-000c29cdf219_2 >

[14] Facultat d'Informàtica de Catalunya. *Distribució de programari* [en línia]. [Consultada: 13 Març 2017]. Disponible a: < <http://www.fib.upc.edu/ca/la-fib/serveis-tic/distribucio-de-programari> >

[15] AngularJS. *AngularJS Developer Guide* [en línia]. [Consultada: 19 Juny 2017]. Disponible a: < <https://docs.angularjs.org/guide> >

[16] AngularJS. *AngularJS Developer Guide - Dependency Injection* [en línia]. [Consultada: 19 Juny 2017]. Disponible a: < <https://docs.angularjs.org/guide/di> >

[17] Spring. *Spring Data JPA - Reference Documentation* [en línia]. [Consultada: 16 Juny 2017].
Disponible a: < <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/#jpa.query-methods.query-creation> >

Annex A: Diagrames de Gantt planificació inicial

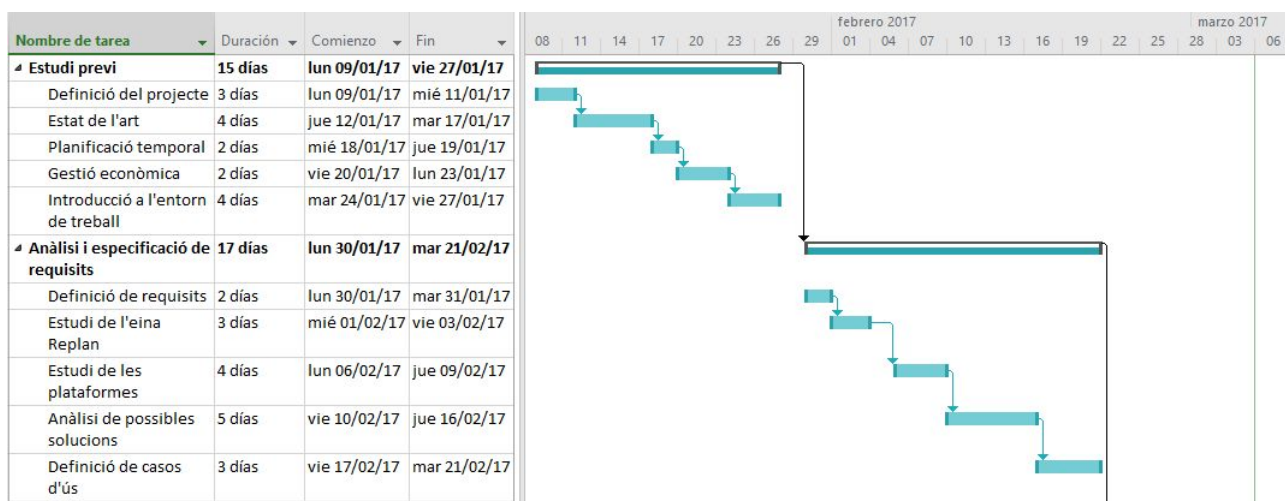


Fig 51. Diagrama Gantt - Fases d'Estudi previ i Anàlisi i especificació de requisits (Font: elaboració pròpia)

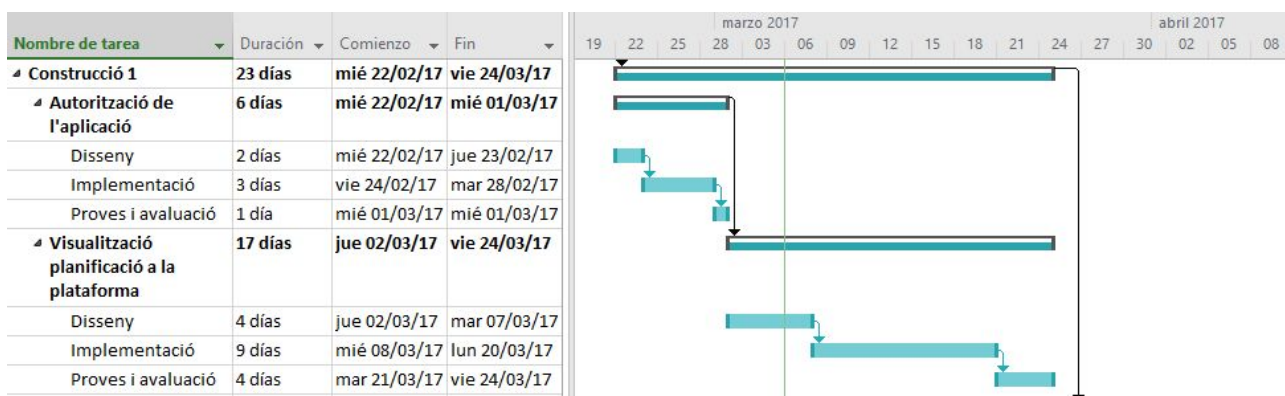


Fig 52. Diagrama Gantt - Fase de Construcció 1 (Font: elaboració pròpia)

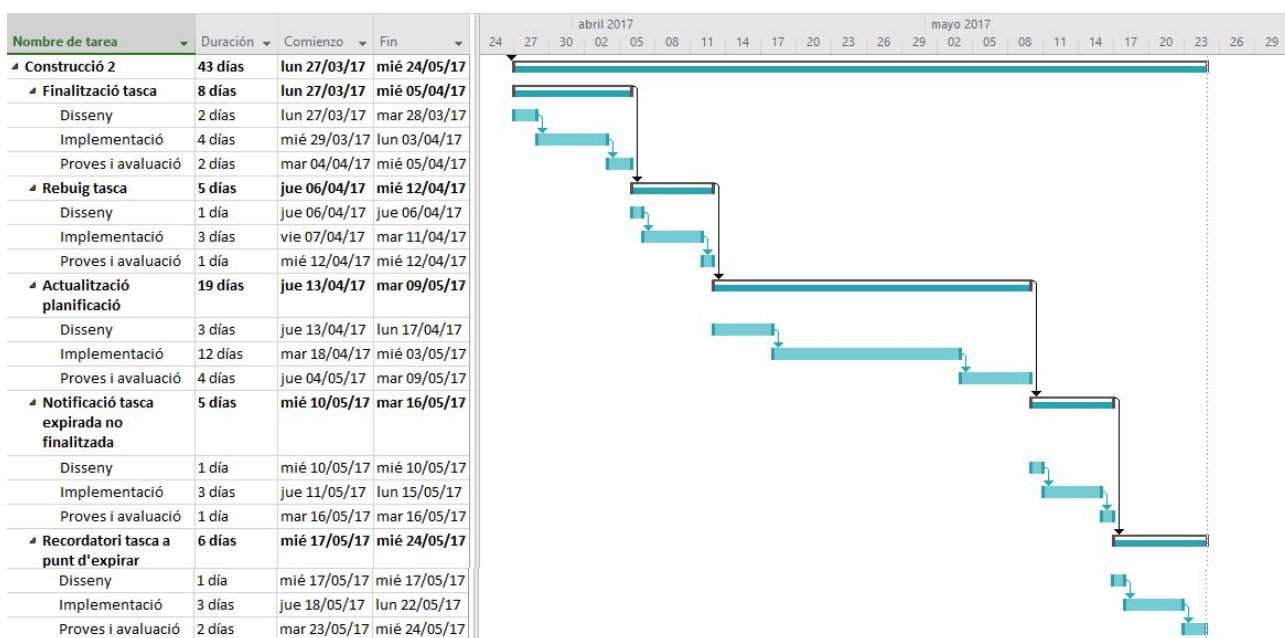


Fig 53. Diagrama Gantt - Fase de Construcció 2 (Font: elaboració pròpia)

Annex B: Diagrames de Gantt final

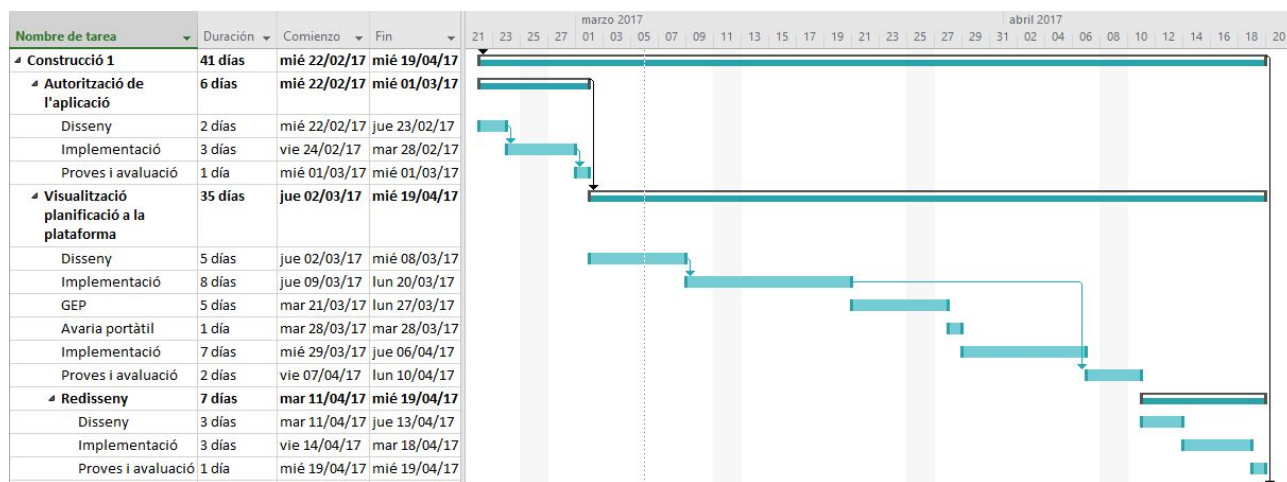


Fig 54. Diagrama Gantt - Fase de Construcció 1 després de les desviacions (Font: elaboració pròpia)

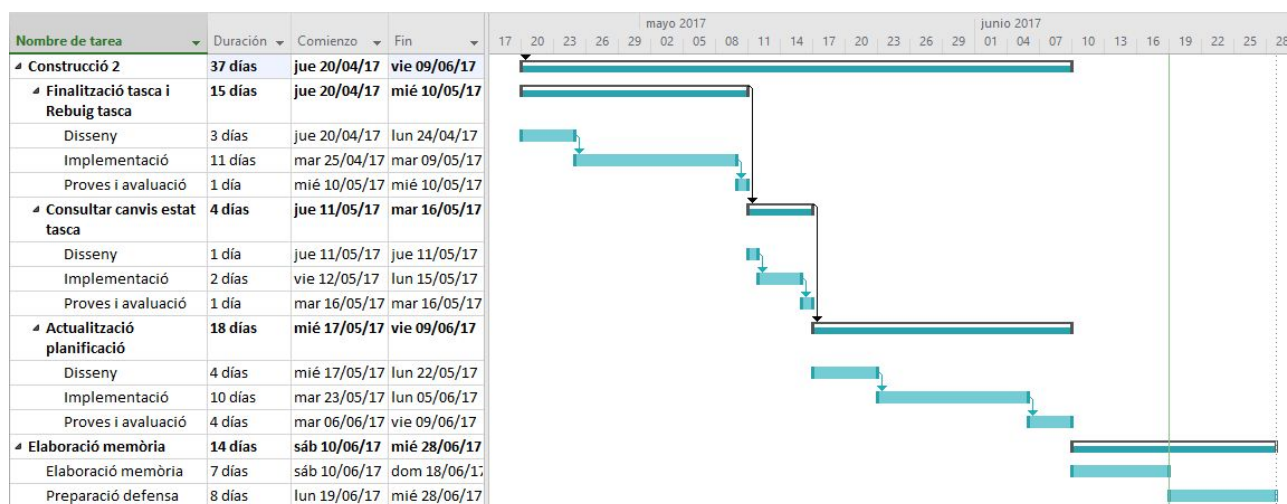


Fig 55. Diagrama Gantt - Fases de Construcció 2 i Elaboració de la memòria després de les desviacions (Font: elaboració pròpia)

Annex C: Especificació de l'API

Replan Plugin for Trello

Replan Plugin for Trello API Documentation

Version: 1.0

BasePath: /

Access

Methods

[[Jump to Models](#)]

Table of Contents

Authenticationcontroller

- [post /authenticate](#)

Boardscontroller

- [post /boards](#)
- [get /boards](#)

Cardscontroller

- [get /card/tracking/{cardId}](#)

Logscontroller

- [post /logs/{logId}/completed](#)
- [post /logs/replan-fake](#)
- [post /logs/replan](#)
- [get /logs](#)
- [post /logs/reject-card](#)

Matchingscontroller

- [post /matchings/create-matchings](#)
- [post /matchings/delete-matchings](#)
- [get /matchings](#)

Projectscontroller

- [get /replan-endpoints](#)
- [get /replan-endpoints/{endpointId}/projects](#)
- [get /replan-endpoints/{endpointId}/projects/{projectId}/releases](#)

Teamscontroller

- get /teams/members
- get /teams

Trellocallbackscontroller

- post /trello-callbacks/cards
- get /trello-callbacks/cards

Userscontroller

- post /users

Authenticationcontroller

Up

post /authenticate

Login (createUserUsingPOST)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

authInfo object (required)

Body Parameter — authInfo (username and password)

Return type

Object

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK Object

401

Unauthorized

404

Not Found

Boardscontroller

Up

post /boards

createBoard (createBoardUsingPOST)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

planBoardDTO PlanBoardDTO (required)

Body Parameter — planBoardDTO

Return type

inline_response_200_1

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK inline_response_200_1

201

Created

404

Not Found

502

Bad Gateway

Up
get /boards
getBoards (getBoardsUsingGET)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Query parameters

username (required)
Query Parameter — username

Return type

array[inline_response_200]

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK

Cardscontroller

Up
get /card/tracking/{cardId}
getCardTrackingInfo (getCardTrackingInfoUsingGET)

Path parameters

cardId (required)
Path Parameter — cardId

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Return type

inline_response_200_2

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses**200**

OK inline_response_200_2

404

Not Found

Logscontroller

Up

post /logs/{logId}/completed

changeAcceptedLog (changeAcceptedLogUsingPOST)

Path parameters

logId (required)

Path Parameter — logId format: int32

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

completeLogOp CompleteLogOp (required)

Body Parameter — completeLogOp

Return type

inline_response_200_3_card

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK inline_response_200_3_card

404

Not Found

Up

post /logs/replan-fake

doReplanFake (doReplanFakeUsingPOST)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

boardId string (required)

Body Parameter — boardId

Return type

array[inline_response_200_1_cards]

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK

Up

post /logs/replan
doReplanProvisional (doReplanProvisionalUsingPOST)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

boardId string (required)
Body Parameter — boardId

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK String

404

Not Found

502

Bad Gateway

Up
get /logs
getLogs (getLogsUsingGET)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Query parameters

username (required)
Query Parameter — username

boardId (optional)

Query Parameter — boardId

Return type

array[inline_response_200_3]

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK

404

Not Found

Up

post /logs/reject-card

rejectCard (rejectCardUsingPOST)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

rejection Rejection (required)

Body Parameter — rejection

Return type

inline_response_200_3

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK inline_response_200_3

404

Not Found

502

Bad Gateway

Matchingscontroller

Up

post /matchings/create-matchings

createMatchings (createMatchingsUsingPOST)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

newMatchings inline_response_200_4_matchings (required)

Body Parameter — newMatchings

Query parameters

username (required)

Query Parameter — username

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

201

Created

404

Not Found

Up
post /matchings/delete-matchings
deleteMatchings (deleteMatchingsUsingPOST)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

matchingsToDelete inline_response_200_4_matchings (required)
Body Parameter — matchingsToDelete

Query parameters

username (required)
Query Parameter — username

Return type

array[inline_response_200_5]

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK

Up
get /matchings
getMatchings (getMatchingsUsingGET)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Query parameters

username (required)
Query Parameter — username

endpointId (required)
Query Parameter — endpointId
projectId (required)
Query Parameter — projectId
releaseId (required)
Query Parameter — releaseId
teamId (required)
Query Parameter — teamId

Return type

inline_response_200_4

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK inline_response_200_4

404

Not Found

Projectscontroller

Up
get /replan-endpoints
getEndpoints (getEndpointsUsingGET)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Return type

array[inline_response_200_6]

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK

Up

get /replan-endpoints/{endpointId}/projects
getProjects (getProjectsUsingGET)

Path parameters

endpointId (required)

Path Parameter — endpointId format: int32

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Return type

array[inline_response_200_7]

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK

404

Not Found

Up

get /replan-endpoints/{endpointId}/projects/{projectId}/releases
getReleases (getReleasesUsingGET)

Path parameters

endpointId (required)

Path Parameter — endpointId format: int32

projectId (required)

Path Parameter — projectId format: int32

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Return type

array[inline_response_200_7]

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK

404

Not Found

Teamscontroller

Up

get /teams/members

getTeamMembers (getTeamMembersUsingGET)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Query parameters

username (required)

Query Parameter — username

teamId (required)

Query Parameter — teamId

Return type

array[inline_response_200_4_member]

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK

404

Not Found

502

Bad Gateway

Up

get /teams

getTeams (getTeamsUsingGET)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Query parameters

username (required)

Query Parameter — username

Return type

array[inline_response_200_8]

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK

502

Bad Gateway

Trellocallbackscontroller

Up

post /trello-callbacks/cards

cardModified (cardModifiedUsingPOST)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

response Response (required)

Body Parameter — response

Query parameters

username (required)

Query Parameter — username

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK String

201

Created

Up

get /trello-callbacks/cards

checkG (checkGUsingGET)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Query parameters

username (required)

Query Parameter — username

Return type

String

Example data

Content-Type:

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

200

OK String

Userscontroller

Up

post /users
createUser (createUserUsingPOST1)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

user WebUser (required)
Body Parameter — user

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- */*

Responses

201

Created

409

Conflict

Up

Models

[[Jump to Methods](#)]

Table of Contents

1. Action
2. Board
3. boards_depends_on
4. boards_feature
5. boards_jobs
6. boards_resource
7. Card
8. CardMovedList
9. CardReduced
10. CardRejection
11. CardTrackingInfo
12. CardTrackingInfoContainer
13. completeLogOp
14. Endpoint

- 15. Feature
- 16. IdNameObject
- 17. inline_response_200
- 18. inline_response_200_1
- 19. inline_response_200_1_cards
- 20. inline_response_200_1_lists
- 21. inline_response_200_2
- 22. inline_response_200_2_trackingInfo
- 23. inline_response_200_3
- 24. inline_response_200_3_card
- 25. inline_response_200_4
- 26. inline_response_200_4_matchings
- 27. inline_response_200_4_member
- 28. inline_response_200_4_plan
- 29. inline_response_200_5
- 30. inline_response_200_6
- 31. inline_response_200_7
- 32. inline_response_200_8
- 33. Job
- 34. JobReduced
- 35. ListTrello
- 36. Log
- 37. Matching
- 38. MatchingsContainer
- 39. Member
- 40. Plan
- 41. planBoardDTO
- 42. PlanTrello
- 43. Project
- 44. rejection
- 45. Release
- 46. Resource
- 47. ResourceMember
- 48. response
- 49. Team
- 50. trelloallbackscards_action
- 51. trelloallbackscards_action_data
- 52. trelloallbackscards_action_data_listAfter
- 53. WebUser
- 54. WebhookCardTrelloResponse

Action - Up

data (optional)

trelloallbackscards_action_data

idMemberCreator (optional)
String
memberCreator (optional)
inline_response_200_4_member
type (optional)
String

Board - Up

id (optional)
String
name (optional)
String
url (optional)
String

boards_depends_on - Up

ends (optional)
String
feature_id (optional)
Integer format: int32
id (optional)
Integer format: int32
resource_id (optional)
Integer format: int32
starts (optional)
String

boards_feature - Up

deadline (optional)
String
description (optional)
String
effort (optional)
Double format: double
id (optional)
Integer format: int32
name (optional)
String

boards_jobs - Up

depends_on (optional)
array[boards_depends_on]
ends (optional)
String

feature (optional)
boards_feature
id (optional)
Integer format: int32
resource (optional)
boards_resource
starts (optional)
String

boards_resource - Up

description (optional)
String
id (optional)
Integer format: int32
name (optional)
String

Card - Up

desc (optional)
String
due (optional)
String
id (optional)
String
idLabels (optional)
array[String]
idList (optional)
String
idMembers (optional)
array[String]
name (optional)
String
pos (optional)
String

CardMovedList - Up

board (optional)
inline_response_200
listAfter (optional)
trellcallbackscards_action_data_listAfter
listBefore (optional)
trellcallbackscards_action_data_listAfter

CardReduced - Up

accepted (optional)

Boolean

alive (optional)

Boolean

boardId (optional)

String

id (optional)

String

name (optional)

String

rejected (optional)

Boolean

CardRejection - Up

boardId (optional)

String

cardId (optional)

String

cardName (optional)

String

comment (optional)

String

username (optional)

String

CardTrackingInfo - Up

createdAt (optional)

String

listName (optional)

String

logId (optional)

Integer format: int32

memberUsername (optional)

String

type (optional)

String

Enum:

FINISHED_EARLIER

FINISHED_LATE

MOVED_TO_IN_PROGRESS

MOVED_TO_READY

REJECTED

CardTrackingInfoContainer - Up

cardId (optional)
String
trackingInfo (optional)
array[inline_response_200_2_trackingInfo]

completeLogOp - Up

accepted (optional)
Boolean

Endpoint - Up

id (optional)
Integer format: int32
name (optional)
String
url (optional)
String

Feature - Up

deadline (optional)
String
description (optional)
String
effort (optional)
Double format: double
id (optional)
Integer format: int32
name (optional)
String

IdNameObject - Up

id (optional)
String
name (optional)
String

inline_response_200 - Up

id (optional)
String
name (optional)
String
url (optional)
String

inline_response_200_1 - Up

board (optional)
inline_response_200
cards (optional)
array[inline_response_200_1_cards]
lists (optional)
array[inline_response_200_1_lists]

inline_response_200_1_cards - Up

desc (optional)
String
due (optional)
String
id (optional)
String
idLabels (optional)
array[String]
idList (optional)
String
idMembers (optional)
array[String]
name (optional)
String
pos (optional)
String

inline_response_200_1_lists - Up

id (optional)
String
idBoard (optional)
String
name (optional)
String

inline_response_200_2 - Up

cardId (optional)
String
trackingInfo (optional)
array[inline_response_200_2_trackingInfo]

inline_response_200_2_trackingInfo - Up

createdAt (optional)
String

listName (optional)

String

logId (optional)

Integer format: int32

memberUsername (optional)

String

type (optional)

String

Enum:

FINISHED_EARLIER

FINISHED_LATE

MOVED_TO_IN_PROGRESS

MOVED_TO_READY

REJECTED

inline_response_200_3 - Up

board (optional)

inline_response_200

card (optional)

inline_response_200_3_card

createdAt (optional)

String

description (optional)

String

id (optional)

Integer format: int32

type (optional)

String

Enum:

FINISHED_EARLIER

FINISHED_LATE

MOVED_TO_IN_PROGRESS

MOVED_TO_READY

REJECTED

inline_response_200_3_card - Up

accepted (optional)

Boolean

alive (optional)

Boolean

boardId (optional)

String

id (optional)

String

name (optional)

String

rejected (optional)

Boolean

inline_response_200_4 - Up

matchings (optional)

array[inline_response_200_4_matchings]

plan (optional)

inline_response_200_4_plan

unmatchedMembers (optional)

array[inline_response_200_4_member]

unmatchedResources (optional)

array[boards_resource]

inline_response_200_4_matchings - Up

member (optional)

inline_response_200_4_member

resource (optional)

boards_resource

inline_response_200_4_member - Up

fullName (optional)

String

id (optional)

String

username (optional)

String

inline_response_200_4_plan - Up

created_at (optional)

String

id (optional)

Integer format: int32

jobs (optional)

array[boards_jobs]

inline_response_200_5 - Up

idResourceMember (optional)

Long format: int64

resourceDescription (optional)

String

resourceId (optional)

Integer format: int32

resourceName (optional)

String

trelloFullName (optional)

String

trelloUserId (optional)

String

trelloUsername (optional)

String

userId (optional)

Long format: int64

inline_response_200_6 - Up

id (optional)

Integer format: int32

name (optional)

String

url (optional)

String

inline_response_200_7 - Up

id (optional)

Integer format: int32

name (optional)

String

inline_response_200_8 - Up

displayName (optional)

String

id (optional)

String

Job - Up

depends_on (optional)

array[boards_depends_on]

ends (optional)

String

feature (optional)

boards_feature

id (optional)

Integer format: int32

resource (optional)

boards_resource

starts (optional)

String

JobReduced - Up

ends (optional)
String
feature_id (optional)
Integer format: int32
id (optional)
Integer format: int32
resource_id (optional)
Integer format: int32
starts (optional)
String

ListTrello - Up

id (optional)
String
idBoard (optional)
String
name (optional)
String

Log - Up

board (optional)
inline_response_200
card (optional)
inline_response_200_3_card
createdAt (optional)
String
description (optional)
String
id (optional)
Integer format: int32
type (optional)
String
Enum:
FINISHED_EARLIER
FINISHED_LATE
MOVED_TO_IN_PROGRESS
MOVED_TO_READY
REJECTED

Matching - Up

member (optional)
inline_response_200_4_member

resource (optional)
boards_resource

MatchingsContainer - Up

matchings (optional)
array[inline_response_200_4_matchings]
plan (optional)
inline_response_200_4_plan
unmatchedMembers (optional)
array[inline_response_200_4_member]
unmatchedResources (optional)
array[boards_resource]

Member - Up

fullName (optional)
String
id (optional)
String
username (optional)
String

Plan - Up

created_at (optional)
String
id (optional)
Integer format: int32
jobs (optional)
array[boards_jobs]

planBoardDTO - Up

boardName (optional)
String
created_at (optional)
String
endpointId (optional)
Integer format: int32
id (optional)
Integer format: int32
jobs (optional)
array[boards_jobs]
projectId (optional)
Integer format: int32
releaseId (optional)
Integer format: int32

teamId (optional)
String
username (optional)
String

PlanTrello - Up

board (optional)
inline_response_200
cards (optional)
array[inline_response_200_1_cards]
lists (optional)
array[inline_response_200_1_lists]

Project - Up

id (optional)
Integer format: int32
name (optional)
String

rejection - Up

boardId (optional)
String
cardId (optional)
String
cardName (optional)
String
comment (optional)
String
username (optional)
String

Release - Up

id (optional)
Integer format: int32
name (optional)
String

Resource - Up

description (optional)
String
id (optional)
Integer format: int32
name (optional)
String

ResourceMember - Up

idResourceMember (optional)
Long format: int64
resourceDescription (optional)
String
resourceId (optional)
Integer format: int32
resourceName (optional)
String
trelloFullName (optional)
String
trelloUserId (optional)
String
trelloUsername (optional)
String
userId (optional)
Long format: int64

response - Up

action (optional)
trellocalbackscards_action
model (optional)
inline_response_200_1_cards

Team - Up

displayName (optional)
String
id (optional)
String

trellocalbackscards_action - Up

data (optional)
trellocalbackscards_action_data
idMemberCreator (optional)
String
memberCreator (optional)
inline_response_200_4_member
type (optional)
String

trellocalbackscards_action_data - Up

board (optional)
inline_response_200

listAfter (optional)
trellcallbackscards_action_data_listAfter
listBefore (optional)
trellcallbackscards_action_data_listAfter

trellcallbackscards_action_data_listAfter - Up

id (optional)
String
name (optional)
String

WebUser - Up

password (optional)
String
trellToken (optional)
String
trellUserId (optional)
String
trellUsername (optional)
String
userId (optional)
Long format: int64
username (optional)
String

WebhookCardTrelloResponse - Up

action (optional)
trellcallbackscards_action
model (optional)
inline_response_200_1_cards

Annex D: Planificació en format JSON 1

```
{
  "id": 1,
  "created_at": "2017-06-17T16:45:49.578Z",
  "release_id": 1,
  "num_features": 7,
  "num_jobs": 7,
  "jobs": [
    {
      "id": 1,
      "starts": "2017-06-01T09:00:00.000Z",
      "ends": "2017-06-08T17:00:00.000Z",
      "feature": {
        "id": 4,
        "code": 444,
        "name": "New channel",
        "description": "Bla, bla, bla",
        "effort": "48.0",
        "deadline": "2017-07-10",
        "priority": 2,
        "release": {
          "release_id": 1
        }
      },
    },
    "resource": {
      "id": 2,
      "name": "Sergi",
      "description": "Sergi Sanchez",
      "availability": "100.0",
      "skills": [
        {
          "id": 2,
          "name": "Ruby",
          "description": "Ruby Programming Language"
        },
        {
          "id": 3,
          "name": "JavaScript",
          "description": "JavaScript Programming Language"
        }
      ]
    },
    "depends_on": []
  },
  {
    "id": 2,
```

```
"starts": "2017-06-01T09:00:00.000Z",
"ends": "2017-06-01T17:00:00.000Z",
"feature": {
  "id": 6,
  "code": 666,
  "name": "Fix user display",
  "description": "Bla, bla, bla",
  "effort": "8.0",
  "deadline": "2017-07-13",
  "priority": 4,
  "release": {
    "release_id": 1
  }
},
"resource": {
  "id": 4,
  "name": "Pere",
  "description": "Pere Prat",
  "availability": "100.0",
  "skills": [
    {
      "id": 1,
      "name": "Java",
      "description": "Java Programming Language"
    },
    {
      "id": 2,
      "name": "Ruby",
      "description": "Ruby Programming Language"
    },
    {
      "id": 3,
      "name": "JavaScript",
      "description": "JavaScript Programming Language"
    }
  ]
},
"depends_on": []
},
{
  "id": 3,
  "starts": "2017-06-01T17:00:00.000Z",
  "ends": "2017-06-02T17:00:00.000Z",
  "feature": {
    "id": 5,
    "code": 555,
    "name": "Email reply",
    "description": "Bla, bla, bla",
    "effort": "8.0",
```

```

"deadline": "2017-07-12",
"priority": 1,
"release": {
  "release_id": 1
},
"resource": {
  "id": 4,
  "name": "Pere",
  "description": "Pere Prat",
  "availability": "100.0",
  "skills": [
    {
      "id": 1,
      "name": "Java",
      "description": "Java Programming Language"
    },
    {
      "id": 2,
      "name": "Ruby",
      "description": "Ruby Programming Language"
    },
    {
      "id": 3,
      "name": "JavaScript",
      "description": "JavaScript Programming Language"
    }
  ]
},
"depends_on": []
},
{
  "id": 4,
  "starts": "2017-06-01T09:00:00.000Z",
  "ends": "2017-06-01T17:00:00.000Z",
  "feature": {
    "id": 1,
    "code": 111,
    "name": "Fix auto upload",
    "description": "Bla, bla, bla",
    "effort": "8.0",
    "deadline": "2017-07-02",
    "priority": 5,
    "release": {
      "release_id": 1
    }
  },
  "resource": {
    "id": 3,

```

```

"name": "Miquel",
"description": "Miquel Munford",
"availability": "100.0",
"skills": [
  {
    "id": 1,
    "name": "Java",
    "description": "Java Programming Language"
  },
  {
    "id": 3,
    "name": "JavaScript",
    "description": "JavaScript Programming Language"
  }
]
},
"depends_on": []
},
{
  "id": 5,
  "starts": "2017-06-01T17:00:00.000Z",
  "ends": "2017-06-02T17:00:00.000Z",
  "feature": {
    "id": 7,
    "code": 777,
    "name": "Profile picture",
    "description": "Bla, bla, bla",
    "effort": "8.0",
    "deadline": "2017-07-15",
    "priority": 5,
    "release": {
      "release_id": 1
    }
  },
  "resource": {
    "id": 3,
    "name": "Miquel",
    "description": "Miquel Munford",
    "availability": "100.0",
    "skills": [
      {
        "id": 1,
        "name": "Java",
        "description": "Java Programming Language"
      },
      {
        "id": 3,
        "name": "JavaScript",
        "description": "JavaScript Programming Language"
      }
    ]
  }
}

```

```

    }
  ]
},
"depends_on": [
  {
    "id": 2,
    "starts": "2017-06-01T09:00:00.000Z",
    "ends": "2017-06-01T17:00:00.000Z",
    "feature_id": 6,
    "resource_id": 4
  }
]
},
{
  "id": 6,
  "starts": "2017-06-01T09:00:00.000Z",
  "ends": "2017-06-05T17:00:00.000Z",
  "feature": {
    "id": 2,
    "code": 222,
    "name": "New login",
    "description": "Bla, bla, bla",
    "effort": "24.0",
    "deadline": "2017-07-06",
    "priority": 4,
    "release": {
      "release_id": 1
    }
  }
},
"resource": {
  "id": 1,
  "name": "Josep",
  "description": "Josep Jackson",
  "availability": "100.0",
  "skills": [
    {
      "id": 1,
      "name": "Java",
      "description": "Java Programming Language"
    },
    {
      "id": 2,
      "name": "Ruby",
      "description": "Ruby Programming Language"
    }
  ]
},
"depends_on": []
},

```



```
{
  "id": 7,
  "starts": "2017-06-05T17:00:00.000Z",
  "ends": "2017-06-08T17:00:00.000Z",
  "feature": {
    "id": 3,
    "code": 333,
    "name": "Enrollment refactoring",
    "description": "Bla, bla, bla",
    "effort": "24.0",
    "deadline": "2017-07-08",
    "priority": 3,
    "release": {
      "release_id": 1
    }
  },
  "resource": {
    "id": 3,
    "name": "Miquel",
    "description": "Miquel Munford",
    "availability": "100.0",
    "skills": [
      {
        "id": 1,
        "name": "Java",
        "description": "Java Programming Language"
      },
      {
        "id": 3,
        "name": "JavaScript",
        "description": "JavaScript Programming Language"
      }
    ]
  },
  "depends_on": [
    {
      "id": 4,
      "starts": "2017-06-01T09:00:00.000Z",
      "ends": "2017-06-01T17:00:00.000Z",
      "feature_id": 1,
      "resource_id": 3
    },
    {
      "id": 6,
      "starts": "2017-06-01T09:00:00.000Z",
      "ends": "2017-06-05T17:00:00.000Z",
      "feature_id": 2,
      "resource_id": 1
    }
  ]
}
```

```
]
}
]
}
```

Annex E: Planificació en format JSON 2

```
{
  "id": 1,
  "created_at": "2017-06-03T15:58:51.673Z",
  "release_id": 5,
  "num_features": 22,
  "num_jobs": 19,
  "jobs": [
    {
      "id": 1,
      "starts": "2017-06-05T09:00:00.000Z",
      "ends": "2017-06-05T13:00:00.000Z",
      "feature": {
        "id": 31,
        "code": 422,
        "name": "Pàgina de contacte",
        "description": "Bla, bla, bla",
        "effort": "4.0",
        "deadline": "2017-07-23",
        "priority": 5,
        "release": {
          "release_id": 5
        }
      }
    },
    {
      "resource": {
        "id": 6,
        "name": "Maria",
        "description": "Maria Miller",
        "availability": "100.0",
        "skills": [
          {
            "id": 4,
            "name": "AngularJS",
            "description": "AngularJS"
          },
          {
            "id": 5,
            "name": "HTML",
            "description": "HyperText Markup Language"
          },
          {
            "id": 6,
            "name": "JavaScript",
            "description": "JavaScript Programming Language"
          },
          {
            "id": 7,
            "name": "CSS",
            "description": "Cascading Stylesheets"
          },
          {
            "id": 9,
```

```

    "name": "SEO",
    "description": "Search Engine Optimization"
  }
],
},
"depends_on": [],
},
{
  "id": 2,
  "starts": "2017-06-05T09:00:00.000Z",
  "ends": "2017-06-05T17:00:00.000Z",
  "feature": {
    "id": 15,
    "code": 406,
    "name": "Maquetació parts comunes",
    "description": "Bla, bla, bla",
    "effort": "8.0",
    "deadline": "2017-07-03",
    "priority": 3,
    "release": {
      "release_id": 5
    }
  },
  "resource": {
    "id": 7,
    "name": "Sergi",
    "description": "Sergi Snow",
    "availability": "100.0",
    "skills": [
      {
        "id": 4,
        "name": "AngularJS",
        "description": "AngularJS"
      },
      {
        "id": 5,
        "name": "HTML",
        "description": "HyperText Markup Language"
      },
      {
        "id": 6,
        "name": "JavaScript",
        "description": "JavaScript Programming Language"
      },
      {
        "id": 7,
        "name": "CSS",
        "description": "Cascading Stylesheets"
      },
      {
        "id": 9,
        "name": "SEO",
        "description": "Search Engine Optimization"
      }
    ]
  }
}

```

```

    },
    "depends_on": []
  },
  {
    "id": 3,
    "starts": "2017-06-05T13:00:00.000Z",
    "ends": "2017-06-05T17:00:00.000Z",
    "feature": {
      "id": 30,
      "code": 421,
      "name": "FAQ",
      "description": "Bla, bla, bla",
      "effort": "4.0",
      "deadline": "2017-07-23",
      "priority": 5,
      "release": {
        "release_id": 5
      }
    },
    "resource": {
      "id": 6,
      "name": "Maria",
      "description": "Maria Miller",
      "availability": "100.0",
      "skills": [
        {
          "id": 4,
          "name": "AngularJS",
          "description": "AngularJS"
        },
        {
          "id": 5,
          "name": "HTML",
          "description": "HyperText Markup Language"
        },
        {
          "id": 6,
          "name": "JavaScript",
          "description": "JavaScript Programming Language"
        },
        {
          "id": 7,
          "name": "CSS",
          "description": "Cascading Stylesheets"
        },
        {
          "id": 9,
          "name": "SEO",
          "description": "Search Engine Optimization"
        }
      ]
    },
    "depends_on": []
  },
  {

```

```
"id": 4,
"starts": "2017-06-05T09:00:00.000Z",
"ends": "2017-06-06T17:00:00.000Z",
"feature": {
  "id": 10,
  "code": 401,
  "name": "Configuració inicial servidor",
  "description": "Bla, bla, bla",
  "effort": "16.0",
  "deadline": "2017-06-09",
  "priority": 1,
  "release": {
    "release_id": 5
  }
},
"resource": {
  "id": 5,
  "name": "Josep",
  "description": "Josep Jackson",
  "availability": "100.0",
  "skills": [
    {
      "id": 10,
      "name": "SQL",
      "description": "Cascading Stylesheets"
    },
    {
      "id": 11,
      "name": "Security",
      "description": "Security"
    },
    {
      "id": 12,
      "name": "Oracle",
      "description": "Oracle Database"
    },
    {
      "id": 13,
      "name": "Ubuntu Server",
      "description": "Ubuntu Server"
    }
  ]
},
"depends_on": []
},
{
  "id": 5,
  "starts": "2017-06-07T09:00:00.000Z",
  "ends": "2017-06-07T17:00:00.000Z",
  "feature": {
    "id": 11,
    "code": 402,
    "name": "Configuració inicial BD",
    "description": "Bla, bla, bla",
    "effort": "8.0",
```

```

"deadline": "2017-06-11",
"priority": 1,
"release": {
  "release_id": 5
},
},
"resource": {
  "id": 5,
  "name": "Josep",
  "description": "Josep Jackson",
  "availability": "100.0",
  "skills": [
    {
      "id": 10,
      "name": "SQL",
      "description": "Cascading Stylesheets"
    },
    {
      "id": 11,
      "name": "Security",
      "description": "Security"
    },
    {
      "id": 12,
      "name": "Oracle",
      "description": "Oracle Database"
    },
    {
      "id": 13,
      "name": "Ubuntu Server",
      "description": "Ubuntu Server"
    }
  ]
},
},
"depends_on": [
  {
    "id": 4,
    "starts": "2017-06-05T09:00:00.000Z",
    "ends": "2017-06-06T17:00:00.000Z",
    "feature_id": 10,
    "resource_id": 5
  }
],
},
{
  "id": 6,
  "starts": "2017-06-08T09:00:00.000Z",
  "ends": "2017-06-09T17:00:00.000Z",
  "feature": {
    "id": 12,
    "code": 403,
    "name": "Disseny BD",
    "description": "Bla, bla, bla",
    "effort": "16.0",
    "deadline": "2017-06-12",

```

```

    "priority": 1,
    "release": {
      "release_id": 5
    }
  },
  "resource": {
    "id": 5,
    "name": "Josep",
    "description": "Josep Jackson",
    "availability": "100.0",
    "skills": [
      {
        "id": 10,
        "name": "SQL",
        "description": "Cascading Stylesheets"
      },
      {
        "id": 11,
        "name": "Security",
        "description": "Security"
      },
      {
        "id": 12,
        "name": "Oracle",
        "description": "Oracle Database"
      },
      {
        "id": 13,
        "name": "Ubuntu Server",
        "description": "Ubuntu Server"
      }
    ]
  },
  "depends_on": [
    {
      "id": 5,
      "starts": "2017-06-07T09:00:00.000Z",
      "ends": "2017-06-07T17:00:00.000Z",
      "feature_id": 11,
      "resource_id": 5
    }
  ]
},
{
  "id": 7,
  "starts": "2017-06-12T09:00:00.000Z",
  "ends": "2017-06-12T13:00:00.000Z",
  "feature": {
    "id": 18,
    "code": 409,
    "name": "Registre",
    "description": "Bla, bla, bla",
    "effort": "4.0",
    "deadline": "2017-06-18",
    "priority": 1,

```



```

    "release": {
      "release_id": 5
    },
    "resource": {
      "id": 9,
      "name": "Albert",
      "description": "Albert Allen",
      "availability": "100.0",
      "skills": [
        {
          "id": 8,
          "name": "PHP",
          "description": "PHP"
        },
        {
          "id": 10,
          "name": "SQL",
          "description": "Cascading Stylesheets"
        }
      ]
    },
    "depends_on": [
      {
        "id": 6,
        "starts": "2017-06-08T09:00:00.000Z",
        "ends": "2017-06-09T17:00:00.000Z",
        "feature_id": 12,
        "resource_id": 5
      }
    ],
    {
      "id": 8,
      "starts": "2017-06-12T09:00:00.000Z",
      "ends": "2017-06-15T17:00:00.000Z",
      "feature": {
        "id": 32,
        "code": 423,
        "name": "Gestió usuaris backend",
        "description": "Bla, bla, bla",
        "effort": "32.0",
        "deadline": "2017-06-19",
        "priority": 1,
        "release": {
          "release_id": 5
        }
      },
      "resource": {
        "id": 5,
        "name": "Josep",
        "description": "Josep Jackson",
        "availability": "100.0",
        "skills": [
          {

```

```

    "id": 10,
    "name": "SQL",
    "description": "Cascading Stylesheets"
  },
  {
    "id": 11,
    "name": "Security",
    "description": "Security"
  },
  {
    "id": 12,
    "name": "Oracle",
    "description": "Oracle Database"
  },
  {
    "id": 13,
    "name": "Ubuntu Server",
    "description": "Ubuntu Server"
  }
]
},
"depends_on": [
  {
    "id": 6,
    "starts": "2017-06-08T09:00:00.000Z",
    "ends": "2017-06-09T17:00:00.000Z",
    "feature_id": 12,
    "resource_id": 5
  }
]
},
{
  "id": 9,
  "starts": "2017-06-12T13:00:00.000Z",
  "ends": "2017-06-12T17:00:00.000Z",
  "feature": {
    "id": 19,
    "code": 410,
    "name": "Login",
    "description": "Bla, bla, bla",
    "effort": "4.0",
    "deadline": "2017-06-18",
    "priority": 1,
    "release": {
      "release_id": 5
    }
  }
},
"resource": {
  "id": 9,
  "name": "Albert",
  "description": "Albert Allen",
  "availability": "100.0",
  "skills": [
    {
      "id": 8,

```

```

    "name": "PHP",
    "description": "PHP"
  },
  {
    "id": 10,
    "name": "SQL",
    "description": "Cascading Stylesheets"
  }
]
},
"depends_on": [
  {
    "id": 6,
    "starts": "2017-06-08T09:00:00.000Z",
    "ends": "2017-06-09T17:00:00.000Z",
    "feature_id": 12,
    "resource_id": 5
  }
]
},
{
  "id": 10,
  "starts": "2017-06-12T09:00:00.000Z",
  "ends": "2017-06-12T17:00:00.000Z",
  "feature": {
    "id": 22,
    "code": 413,
    "name": "Obtenir productes",
    "description": "Bla, bla, bla",
    "effort": "8.0",
    "deadline": "2017-06-28",
    "priority": 2,
    "release": {
      "release_id": 5
    }
  }
},
"resource": {
  "id": 8,
  "name": "Pere",
  "description": "Pere Patterson",
  "availability": "100.0",
  "skills": [
    {
      "id": 8,
      "name": "PHP",
      "description": "PHP"
    },
    {
      "id": 10,
      "name": "SQL",
      "description": "Cascading Stylesheets"
    }
  ]
},
"depends_on": [

```

```
{
  "id": 6,
  "starts": "2017-06-08T09:00:00.000Z",
  "ends": "2017-06-09T17:00:00.000Z",
  "feature_id": 12,
  "resource_id": 5
}
],
{
  "id": 11,
  "starts": "2017-06-13T09:00:00.000Z",
  "ends": "2017-06-13T17:00:00.000Z",
  "feature": {
    "id": 33,
    "code": 424,
    "name": "Control sessions",
    "description": "Bla, bla, bla",
    "effort": "8.0",
    "deadline": "2017-06-20",
    "priority": 1,
    "release": {
      "release_id": 5
    }
  },
  "resource": {
    "id": 9,
    "name": "Albert",
    "description": "Albert Allen",
    "availability": "100.0",
    "skills": [
      {
        "id": 8,
        "name": "PHP",
        "description": "PHP"
      },
      {
        "id": 10,
        "name": "SQL",
        "description": "Cascading Stylesheets"
      }
    ]
  }
},
"depends_on": [
  {
    "id": 6,
    "starts": "2017-06-08T09:00:00.000Z",
    "ends": "2017-06-09T17:00:00.000Z",
    "feature_id": 12,
    "resource_id": 5
  },
  {
    "id": 9,
    "starts": "2017-06-12T13:00:00.000Z",
    "ends": "2017-06-12T17:00:00.000Z",
```

```

    "feature_id": 19,
    "resource_id": 9
  }
],
{
  "id": 12,
  "starts": "2017-06-13T09:00:00.000Z",
  "ends": "2017-06-15T17:00:00.000Z",
  "feature": {
    "id": 23,
    "code": 414,
    "name": "Cercar productes",
    "description": "Bla, bla, bla",
    "effort": "24.0",
    "deadline": "2017-06-30",
    "priority": 2,
    "release": {
      "release_id": 5
    }
  },
  "resource": {
    "id": 8,
    "name": "Pere",
    "description": "Pere Patterson",
    "availability": "100.0",
    "skills": [
      {
        "id": 8,
        "name": "PHP",
        "description": "PHP"
      },
      {
        "id": 10,
        "name": "SQL",
        "description": "Cascading Stylesheets"
      }
    ]
  },
  "depends_on": [
    {
      "id": 10,
      "starts": "2017-06-12T09:00:00.000Z",
      "ends": "2017-06-12T17:00:00.000Z",
      "feature_id": 22,
      "resource_id": 8
    }
  ]
},
{
  "id": 13,
  "starts": "2017-06-13T09:00:00.000Z",
  "ends": "2017-06-13T17:00:00.000Z",
  "feature": {
    "id": 13,

```

```

"code": 404,
"name": "Pàgina registre",
"description": "Bla, bla, bla",
"effort": "8.0",
"deadline": "2017-06-23",
"priority": 1,
"release": {
  "release_id": 5
},
},
"resource": {
  "id": 6,
  "name": "Maria",
  "description": "Maria Miller",
  "availability": "100.0",
  "skills": [
    {
      "id": 4,
      "name": "AngularJS",
      "description": "AngularJS"
    },
    {
      "id": 5,
      "name": "HTML",
      "description": "HyperText Markup Language"
    },
    {
      "id": 6,
      "name": "JavaScript",
      "description": "JavaScript Programming Language"
    },
    {
      "id": 7,
      "name": "CSS",
      "description": "Cascading Stylesheets"
    },
    {
      "id": 9,
      "name": "SEO",
      "description": "Search Engine Optimization"
    }
  ]
},
"depends_on": [
  {
    "id": 7,
    "starts": "2017-06-10T09:00:00.000Z",
    "ends": "2017-06-12T13:00:00.000Z",
    "feature_id": 18,
    "resource_id": 9
  }
],
},
{
  "id": 14,

```

```
"starts": "2017-06-14T09:00:00.000Z",
"ends": "2017-06-14T17:00:00.000Z",
"feature": {
  "id": 26,
  "code": 417,
  "name": "Obtenir informació usuari",
  "description": "Bla, bla, bla",
  "effort": "8.0",
  "deadline": "2017-07-08",
  "priority": 3,
  "release": {
    "release_id": 5
  }
},
"resource": {
  "id": 9,
  "name": "Albert",
  "description": "Albert Allen",
  "availability": "100.0",
  "skills": [
    {
      "id": 8,
      "name": "PHP",
      "description": "PHP"
    },
    {
      "id": 10,
      "name": "SQL",
      "description": "Cascading Stylesheets"
    }
  ]
},
"depends_on": [
  {
    "id": 6,
    "starts": "2017-06-08T09:00:00.000Z",
    "ends": "2017-06-09T17:00:00.000Z",
    "feature_id": 12,
    "resource_id": 5
  }
],
{
  "id": 15,
  "starts": "2017-06-14T09:00:00.000Z",
  "ends": "2017-06-14T17:00:00.000Z",
  "feature": {
    "id": 14,
    "code": 405,
    "name": "Pàgina login",
    "description": "Bla, bla, bla",
    "effort": "8.0",
    "deadline": "2017-06-23",
    "priority": 1,
    "release": {
```

```

    "release_id": 5
  },
  "resource": {
    "id": 6,
    "name": "Maria",
    "description": "Maria Miller",
    "availability": "100.0",
    "skills": [
      {
        "id": 4,
        "name": "AngularJS",
        "description": "AngularJS"
      },
      {
        "id": 5,
        "name": "HTML",
        "description": "HyperText Markup Language"
      },
      {
        "id": 6,
        "name": "JavaScript",
        "description": "JavaScript Programming Language"
      },
      {
        "id": 7,
        "name": "CSS",
        "description": "Cascading Stylesheets"
      },
      {
        "id": 9,
        "name": "SEO",
        "description": "Search Engine Optimization"
      }
    ]
  },
  "depends_on": [
    {
      "id": 9,
      "starts": "2017-06-12T13:00:00.000Z",
      "ends": "2017-06-12T17:00:00.000Z",
      "feature_id": 19,
      "resource_id": 9
    }
  ],
  {
    "id": 16,
    "starts": "2017-06-15T09:00:00.000Z",
    "ends": "2017-06-15T17:00:00.000Z",
    "feature": {
      "id": 29,
      "code": 420,
      "name": "Modificar informació usuari",
      "description": "Bla, bla, bla",

```



```

    "effort": "8.0",
    "deadline": "2017-07-01",
    "priority": 3,
    "release": {
      "release_id": 5
    }
  },
  "resource": {
    "id": 9,
    "name": "Albert",
    "description": "Albert Allen",
    "availability": "100.0",
    "skills": [
      {
        "id": 8,
        "name": "PHP",
        "description": "PHP"
      },
      {
        "id": 10,
        "name": "SQL",
        "description": "Cascading Stylesheets"
      }
    ]
  },
  "depends_on": [
    {
      "id": 6,
      "starts": "2017-06-08T09:00:00.000Z",
      "ends": "2017-06-09T17:00:00.000Z",
      "feature_id": 12,
      "resource_id": 5
    }
  ],
  {
    "id": 17,
    "starts": "2017-06-16T09:00:00.000Z",
    "ends": "2017-06-16T17:00:00.000Z",
    "feature": {
      "id": 28,
      "code": 419,
      "name": "Canviar contrasenya",
      "description": "Bla, bla, bla",
      "effort": "8.0",
      "deadline": "2017-07-01",
      "priority": 3,
      "release": {
        "release_id": 5
      }
    }
  },
  "resource": {
    "id": 9,
    "name": "Albert",
    "description": "Albert Allen",

```

```
"availability": "100.0",
"skills": [
  {
    "id": 8,
    "name": "PHP",
    "description": "PHP"
  },
  {
    "id": 10,
    "name": "SQL",
    "description": "Cascading Stylesheets"
  }
],
"depends_on": [
  {
    "id": 6,
    "starts": "2017-06-08T09:00:00.000Z",
    "ends": "2017-06-09T17:00:00.000Z",
    "feature_id": 12,
    "resource_id": 5
  }
],
},
{
  "id": 18,
  "starts": "2017-06-16T09:00:00.000Z",
  "ends": "2017-06-22T17:00:00.000Z",
  "feature": {
    "id": 24,
    "code": 415,
    "name": "Comprar productes",
    "description": "Bla, bla, bla",
    "effort": "16.0",
    "deadline": "2017-07-03",
    "priority": 2,
    "release": {
      "release_id": 5
    }
  },
  "resource": {
    "id": 8,
    "name": "Pere",
    "description": "Pere Patterson",
    "availability": "100.0",
    "skills": [
      {
        "id": 8,
        "name": "PHP",
        "description": "PHP"
      },
      {
        "id": 10,
        "name": "SQL",
        "description": "Cascading Stylesheets"
      }
    ]
  }
}
```

```

    }
  ]
},
"depends_on": [
  {
    "id": 6,
    "starts": "2017-06-08T09:00:00.000Z",
    "ends": "2017-06-09T17:00:00.000Z",
    "feature_id": 12,
    "resource_id": 5
  }
]
},
{
  "id": 19,
  "starts": "2017-06-16T09:00:00.000Z",
  "ends": "2017-06-21T17:00:00.000Z",
  "feature": {
    "id": 17,
    "code": 408,
    "name": "Pàgina home",
    "description": "Bla, bla, bla",
    "effort": "32.0",
    "deadline": "2017-07-02",
    "priority": 2,
    "release": {
      "release_id": 5
    }
  }
},
"resource": {
  "id": 6,
  "name": "Maria",
  "description": "Maria Miller",
  "availability": "100.0",
  "skills": [
    {
      "id": 4,
      "name": "AngularJS",
      "description": "AngularJS"
    },
    {
      "id": 5,
      "name": "HTML",
      "description": "HyperText Markup Language"
    },
    {
      "id": 6,
      "name": "JavaScript",
      "description": "JavaScript Programming Language"
    },
    {
      "id": 7,
      "name": "CSS",
      "description": "Cascading Stylesheets"
    }
  ],

```

```

    {
      "id": 9,
      "name": "SEO",
      "description": "Search Engine Optimization"
    }
  ],
},
"depends_on": [
  {
    "id": 12,
    "starts": "2017-06-13T09:00:00.000Z",
    "ends": "2017-06-15T17:00:00.000Z",
    "feature_id": 23,
    "resource_id": 8
  }
]
},
{
  "id": 20,
  "starts": "2017-06-16T09:00:00.000Z",
  "ends": "2017-06-23T17:00:00.000Z",
  "feature": {
    "id": 16,
    "code": 407,
    "name": "Pàgina producte",
    "description": "Bla, bla, bla",
    "effort": "56.0",
    "deadline": "2017-06-30",
    "priority": 2,
    "release": {
      "release_id": 5
    }
  },
  "resource": {
    "id": 7,
    "name": "Sergi",
    "description": "Sergi Snow",
    "availability": "100.0",
    "skills": [
      {
        "id": 4,
        "name": "AngularJS",
        "description": "AngularJS"
      },
      {
        "id": 5,
        "name": "HTML",
        "description": "HyperText Markup Language"
      },
      {
        "id": 6,
        "name": "JavaScript",
        "description": "JavaScript Programming Language"
      }
    ]
  }
}

```

```

    "id": 7,
    "name": "CSS",
    "description": "Cascading Stylesheets"
  },
  {
    "id": 9,
    "name": "SEO",
    "description": "Search Engine Optimization"
  }
]
},
"depends_on": [
  {
    "id": 12,
    "starts": "2017-06-13T09:00:00.000Z",
    "ends": "2017-06-15T17:00:00.000Z",
    "feature_id": 23,
    "resource_id": 8
  }
]
},
{
  "id": 21,
  "starts": "2017-06-19T09:00:00.000Z",
  "ends": "2017-06-23T17:00:00.000Z",
  "feature": {
    "id": 21,
    "code": 412,
    "name": "Gestió inventari",
    "description": "Bla, bla, bla",
    "effort": "40.0",
    "deadline": "2017-07-01",
    "priority": 4,
    "release": {
      "release_id": 5
    }
  }
},
"resource": {
  "id": 9,
  "name": "Albert",
  "description": "Albert Allen",
  "availability": "100.0",
  "skills": [
    {
      "id": 8,
      "name": "PHP",
      "description": "PHP"
    },
    {
      "id": 10,
      "name": "SQL",
      "description": "Cascading Stylesheets"
    }
  ]
}
},

```

```
"depends_on": [
  {
    "id": 8,
    "starts": "2017-06-12T09:00:00.000Z",
    "ends": "2017-06-15T17:00:00.000Z",
    "feature_id": 32,
    "resource_id": 5
  }
],
{
  "id": 22,
  "starts": "2017-06-22T09:00:00.000Z",
  "ends": "2017-06-23T17:00:00.000Z",
  "feature": {
    "id": 27,
    "code": 418,
    "name": "Pàgina informació usuari",
    "description": "Bla, bla, bla",
    "effort": "16.0",
    "deadline": "2017-07-01",
    "priority": 3,
    "release": {
      "release_id": 5
    }
  },
  "resource": {
    "id": 6,
    "name": "Maria",
    "description": "Maria Miller",
    "availability": "100.0",
    "skills": [
      {
        "id": 4,
        "name": "AngularJS",
        "description": "AngularJS"
      },
      {
        "id": 5,
        "name": "HTML",
        "description": "HyperText Markup Language"
      },
      {
        "id": 6,
        "name": "JavaScript",
        "description": "JavaScript Programming Language"
      },
      {
        "id": 7,
        "name": "CSS",
        "description": "Cascading Stylesheets"
      },
      {
        "id": 9,
        "name": "SEO",
```

```

    "description": "Search Engine Optimization"
  }
]
},
"depends_on": [
  {
    "id": 14,
    "starts": "2017-06-14T09:00:00.000Z",
    "ends": "2017-06-14T17:00:00.000Z",
    "feature_id": 26,
    "resource_id": 9
  },
  {
    "id": 17,
    "starts": "2017-06-16T09:00:00.000Z",
    "ends": "2017-06-16T17:00:00.000Z",
    "feature_id": 28,
    "resource_id": 9
  },
  {
    "id": 16,
    "starts": "2017-06-15T09:00:00.000Z",
    "ends": "2017-06-15T17:00:00.000Z",
    "feature_id": 29,
    "resource_id": 9
  }
]
}
]
}

```